R86-21

206p.

IN-34309

MIT

SPATIAL ANALYSIS OF STORM DEPTHS FROM AN ARIZONA RAINGAGE NETWORK

by
NEIL M. FENNESSEY
PETER S. EAGLESON
WANG QINLIANG
and
IGNACIO RODRIGUEZ-ITURBE

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Report No. 306

Prepared with support of

The National Aeronautics and Space Administration
Grant No. NAG 5-388
and
The National Science Foundation
Grant No. ATM 8420781

DEPARTMENT
OF
CIVIL
ENGINEERING

SCHOOL OF ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOG
Cambridge, Massachusetts 02139

R86-21

SPATIAL ANALYSIS OF STORM DEPTHS FROM AN ARIZONA RAINGAGE NETWORK

by

Neil M. Fennessey
Peter S. Eagleson
Wang Qinliang
and
Ignacio Rodriguez-Iturbe

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Department of Civil Engineering
Massachusetts Institute of Technology

Report No. 306

Prepared with the support of

The National Aeronautics and Space Administration
Grant No. NAG 5-388

and

The National Science Foundation
Grant No. ATM 8420781

August, 1986

PREFACE

This report is one of four in a sequence discribing and evaluating the modeling of the total storm rainfall due to stationary events. These reports are:

Report No. 305 "Spatial Poisson Models of Stationary Storm Rainfall: Theoretical Development" by I. Rodriguez-Iturbe, Q. Wang, P. S. Eagleson and B. L. Jacobs.

Report No. 306 "Spatial Analysis of Storm Depths from an Arizona Raingage Network" by N. M. Fennessey, P. S. Eagleson, Q. Wang and I. Rodriguez-Iturbe.

Report No. 307 (Vols. 1 and 2) "Spatial Characteristics of Observed Precipitation Fields: A Catalog of Summer Storms in Arizona" by N. M. Fennessey, P. S. Eagleson, Q. Wang and I. Rodriguez-Iturbe.

Report No. 308 "Spatial Poisson Models of Stationary Storm Rainfall: Parameterization, Evaluation and Numerical Simulation" by N. M. Fennessey, Q. Wang, P. S. Eagleson and I. Rodriguez-Iturbe.

They are all available from

Director
Ralph M. Parsons Laboratory
Room 48-311
Massachusetts Institute of Technology
Cambridge, MA 02139

The raw data were provided by the Agricultural Research Service (U.S. Department of Agriculture), and are available from their data center in Beltsville, Maryland.

Data tables for the 428 identified storms are available on computer tape from the above MIT address, along with a computer program for retrieving the data for a particular storm.

ABSTRACT

Eight years of summer rainstorm observations are analyzed by a dense network of 93 raingages operated by the U. S. Department of Agriculture, Agricultural Research Service, in the 150 km Walnut Gulch experimental catchment near Tucson, Arizona. Storms are defined by the total depths collected at each raingage during the noon-to-noon period for which there was depth recorded at any of the gages. For each of the resulting 428 storm days, the 93 gage depths are interpolated onto a dense grid and the resulting random field analyzed to obtain moments, isohyetal plots, spatial correlation function, variance function, and the spatial distribution of storm depth.

ACKNOWLEDGMENTS

This study was supported (serially) by the National Aeronautics and Space Administration under Grant No. NAG 5-388 and by the National Science Foundation under Grant No. ATM-8420781.

The research was performed by Mr. Neil M. Fennessey, Research Assistant in Civil Engineering, and this document is a part of his thesis submitted to the Massachusetts Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering. The work was supervised by Dr. Peter S. Eagleson, Edmund K. Turner Professor of Civil Engineering. Mr. Wang Qinliang, Visiting Engineer, provided continuing assistance while on leave from the Yangtze Valley Planning Office, Ministry of Water Resources and Electric Power, People's Republic of China. Dr. Ignacio Rodriguez-Iturbe, Graduate Program in Hydrology and Water Resources, Universidad Simon Bolivar, Caracas, Venezuela, provided guidance on a consulting basis.

Special thanks are extended to Dr. David Woolhiser and to Ms. Fatima Lopez of the U.S.D.A. Southwest Rangeland Watershed Research Center in Tucson, Arizona for generously providing the data used in this study.

TABLE OF CONTENTS

List of Figures

List of Tables

Notation

A	area, km^2
A_c	catchment area, km^2
A_{cd}	dry area of catchment, km^2
A_{cw}	wetted area of catchment
A_{cd}/A_c	dry fraction of catchment area
A_{cw}/A_c	wetted fraction of catchment area
A_s	storm area, km^2
A_{sc}	storm-catchment overlap area, km^2
C.S.	coefficient of skewness
$F_Y(y)$	cumulative probability function of $Y(\underline{x})$
$I_i(\ )$	ith order modified Bessel function of the first kind
$K_i(\ )$	ith order modified Bessel function of the second kind.
L	length, km
$L_i(\ )$	ith order Struve function
R_2	2-dimensional plane
R_k	spatial Poisson "nearest neighbor distance," km, simulation distance, km
R_L	maximum simulation distance, km
R_o	maximum simulation distance, km
Y	point storm depth, mm
Y_A	variance function element storm depth, mm
Y_k	finite element storm depth, mm
Y_o	station observation storm depth, mm
a	storm cell spread function parameter, km^{-1} or dimensionless

Notation

(Continued)

erf( )	error function
$f(a)$	standard square error function of parameter a
$f(\alpha)$	probability density function of storm cell parameter α
$f_{R_k}(r)$	probability density function of the spatial Poisson "nearest neighbor distances"
$f_t(\ )$	"temporal" probability density function
$f_Y(y)$	probability density function of $Y(\underline{x})$
$g(\underline{x},\underline{z},\alpha)$	storm cell depth spread function
nR	number of simulation storm cells
q	bivariate surface interpolator polynomial coefficient
r	radial distance, km
u	complex variable
$\underline{x}$	2-dimensional position vector
$\bar{x}$	expected value of the node depth, mm
$\bar{\bar{x}}$	expected value of the variance function element depth, mm
y	point storm depth, mm
$\underline{z}$	2-dimensional position vector
α	storm cell center depth parameter, mm
$\hat{\hat{\alpha}}$	maximum simulation distance parameter
β	exponential distribution parameter, mm^{-1} or dimensionless
$\gamma[A]$	variance function at area A
$\delta(\ )$	delta Dirac function
δ	gamma distribution parameter

Notation

(Continued)

ε	exponential distribution parameter
θ	gamma distribution parameter
κ	gamma distribution parameter
λ	storm cell density parameter, km^{-2}
ν	lag distance, km
$\rho(\nu)$	spatial correlation coefficient at lag ν
$\rho_t(\nu)$	theoretical spatial correlation coefficient
σ_A	standard deviation of $Y(\underline{x})$ averaged over area A, mm
σ_k	standard deviation of the finite element storm depth, mm
σ_Y	standard deviation of $Y(\underline{x})$
$\phi_Y(\ )$	characteristic function
χ	gamma distribution parameter

CHAPTER 1

Introduction

1.1 Background

Current atmospheric general circulation models (GCMs) use complex numerical techniques to solve the hydrodynamic and thermodynamic equations for the atmosphere, but they generally treat the landsurface moisture boundary condition in a rather simplistic way. An important improvement upon existing GCMs would result from incorporation of the subgrid-scale spatial characteristics of storm rainfall.

There are many problems inherent in a spatial analysis of rainfall. First, there is a lack of data collection networks of sufficiently small resolution and of areal extent approaching the scale (10^4 km^2) of a typical GCM grid square. This is primarily due to the operational costs of such networks. The difference in scale between a single GCM grid square and the few existing dense raingage networks is typically at least one order of magnitude.

Related to the adequacy of the spatial scale of resolution of raingage networks is the broad variation among recognized precipitation systems. It is well known that different types of precipitation systems may span several orders of magnitude both spatially and temporally. Orlanski [1975] summarized the scales of various types of precipitation systems. Thunderstorms have characteristic scales between 2-20 km spatially and between two minutes and four hours temporally. Frontal systems and hurricanes may range from 200-2000 km spatially and from one day to a week temporally. Others have reported that most of the mid-latitude precipitation occurs in storms that are on the order of

20-2000 km in scale. Clearly, dense data collection networks cannot be operated with sufficient resolution to encompass such large scale storm systems. Another important issue to consider with respect to a spatial analysis of precipitation is the movement or translation of the event or components of the event. The movement of large scale frontal systems has been widely investigated. Rainbands located within cyclonic frontal systems may move at rates of 60-70 kilometers per hour. Such a system of characteristic spatial scale and duration will move quickly into and out of the typical dense raingage network. In contrast, convective cells within a local thundershower may move at most a few kilometers an hour during the lifetime of the event.

For these reasons, we focus on rainstorms of limited areal extent and limited translation, specifically, the air-mass thunderstorm. The storms studied were observed by the dense raingage network maintained by the Agricultural Research Service (USDA) in the Walnut Gulch, Arizona experimental catchment.

1.2 Goals and Objectives

The general objective of this study is to perform a probabilistic analysis of the spatial random field of total rainfall depth resulting from stationary thunderstorms with a view toward developing general descriptions of the spatial characteristics of such storms.

The specific objective of this report is:

(1) to use station observations from a dense raingage network to estimate the moments, spatial correlation, variance of the local averaging process, and the spatial distribution of total storm rainfall depth.

CHAPTER 2

General Characteristics of Air-Mass Thunderstorms

2.1 Air-Mass Thunderstorms

Thunderstorm rainfall is one type of precipitation associated with air-mass movement. As a general rule, cyclonic action is weak during the summer months in the mid-latitudes, and the predominant type of event is the thunderstorm. Several types of thunderstorms are recognized and are distinguished by the actions of the air mass in which the event occurs. The principal categories include air-mass thunderstorms, frontal thunderstorms (associated with either a warm or a cold front) and squall line thunderstorms.

Battan [1984] defines an air-mass as a widespread body of air that is approximately homogeneous in its horizonal extent, particularly with respect to its distribution of temperature, moisture and their respective vertical gradients. An air-mass thunderstorm occurs when the moist air within the air-mass becomes unstable.

The air-mass thunderstorm typically results from intensive daytime heating of the land surface. This surface heating creates a conditional instability (see Cole [1980]), which may lead to the development of local convective circulation. Because of the surface origin of the convective disturbance these storms tend to be relatively stationary.

In its early stage of development, the air-mass thunderstorm consists primarily of one or more actively rising thermals. These convective cells cool as they rise to higher elevations; the local moist air attains saturation and moisture condenses out as raindrops. Entrainment of subsaturated air from outside the convective cell further contributes to the cessation of these updrafts, and the vertical buoyancy of the convective cell is arrested.

At this stage, evaporation of raindrops leads to local cooling and to the development of downdrafts, which carry precipitation and cool air to the surface. Updrafts and downdrafts may co-exist side by. Eventually as the storm matures, the convective updrafts weaken from a lack of local warm air available for further entrainment and the storm begins to dissipate. Once the air-mass thunderstorm has overturned the local atmosphere, static stability is restored. Characteristically, air-mass thunderstorms occur in the mid- to late afternoon when land surface temperatures are at their greatest. The duration of these events tend to be on the order of an hour.

2.2 Storm Movement

One of the objectives of this study involves the modeling of total rainfall depth generated by assumed stationary convective cells and is reported in a companion volume. For this purpose, we shall assume that the air-mass thunderstorms in the area of study are horizontally motionless in space. This assumption is supported by the work of Fletcher [1960] who found that air-mass thunderstorms in Arizona have little translational motion.

CHAPTER 3

The Walnut Gulch Experimental Watershed

3.1 Selection of the Data Network

In 1953, two southwest rangeland watersheds were selected by the Agricultural Research Service (U.S.D.A.) as field laboratories [Osborn et al., 1979]. The primary objective of the envisioned field research was to quantify water yield from rangeland watersheds. The first watershed chosen was the 150 km^2 Walnut Gulch basin located in southeastern Arizona. The second watershed was the 174 km^2 Alamogordo Creek basin located in eastern New Mexico.

According to Osborn [1967], the Walnut Gulch watershed represents a region in the southwest U.S. where almost all the thunderstorms result from purely surface heating, e.g., air-mass thunderstorms. He states that in contrast, the Alamogordo Creek watershed is representative of a region where thunderstorms form from combined surface heating and frontal activity, as well as by purely surface heating.

Osborn and Hickok [1968] determined that 70% of the annual rainfall and essentially all of the runoff from rangelands in southeastern Arizona result from air-mass thunderstorms. Osborn and Laursen [1973] found that despite the frequency of air-mass thunderstorms, the more massive frontal convective thunderstorms contribute the higher proportion of annual rainfall and subsequent runoff in eastern New Mexico. They found that frontal convective thunderstorms are essentially non-existent in Walnut Gulch during the summer months.

Because the objective of this study is to focus on stationary thunderstorms and because frontal system thunderstorms are usually moving, the Walnut Gulch experimental watershed is selected for analysis. We will assume that all summer storms on the watershed are air-mass thunderstorms.

3.2 Geographical Characteristics of the Walnut Gulch Watershed

Walnut Gulch is an ephemeral stream located in the San Pedro River basin, in southeastern Arizona. The Walnut Gulch basin is similar to much of the brush-grass rangeland found elsewhere in the southwestern United States in that the lower two thirds of the basin is largely brush covered, and the remaining higher reaches are covered by mostly native grasses [Osborn and Laursen, 1973]. The elevation of the watershed ranges from 4000 to 6000 feet as shown in Figure 3.2.1, where the elevation contours are drawn at 200 foot intervals.

3.3 Summertime Rainstorm Characteristics

Renard and Brakensiek [1976] report that the characteristics of precipitation in the rangeland areas of the United States are varied and depend on the atmospheric moisture source, the season, and the elevation among other things. Battan [1984] states that much of the winter moisture in the western rangelands comes from the Pacific Ocean, carried into the region by westerly winds. There has been a great deal of debate as to whether the moisture source of the summertime thunderstorm is the Pacific Ocean or the Gulf of Mexico (see Osborn [1967]; Osborn and Lane [1972]; Hales [1973]; Osborn and Davis [1977]). More recently, however, using GCM experiments, Koster et al. [1986] show that the predominant summertime moisture source of this region is the continental U.S.

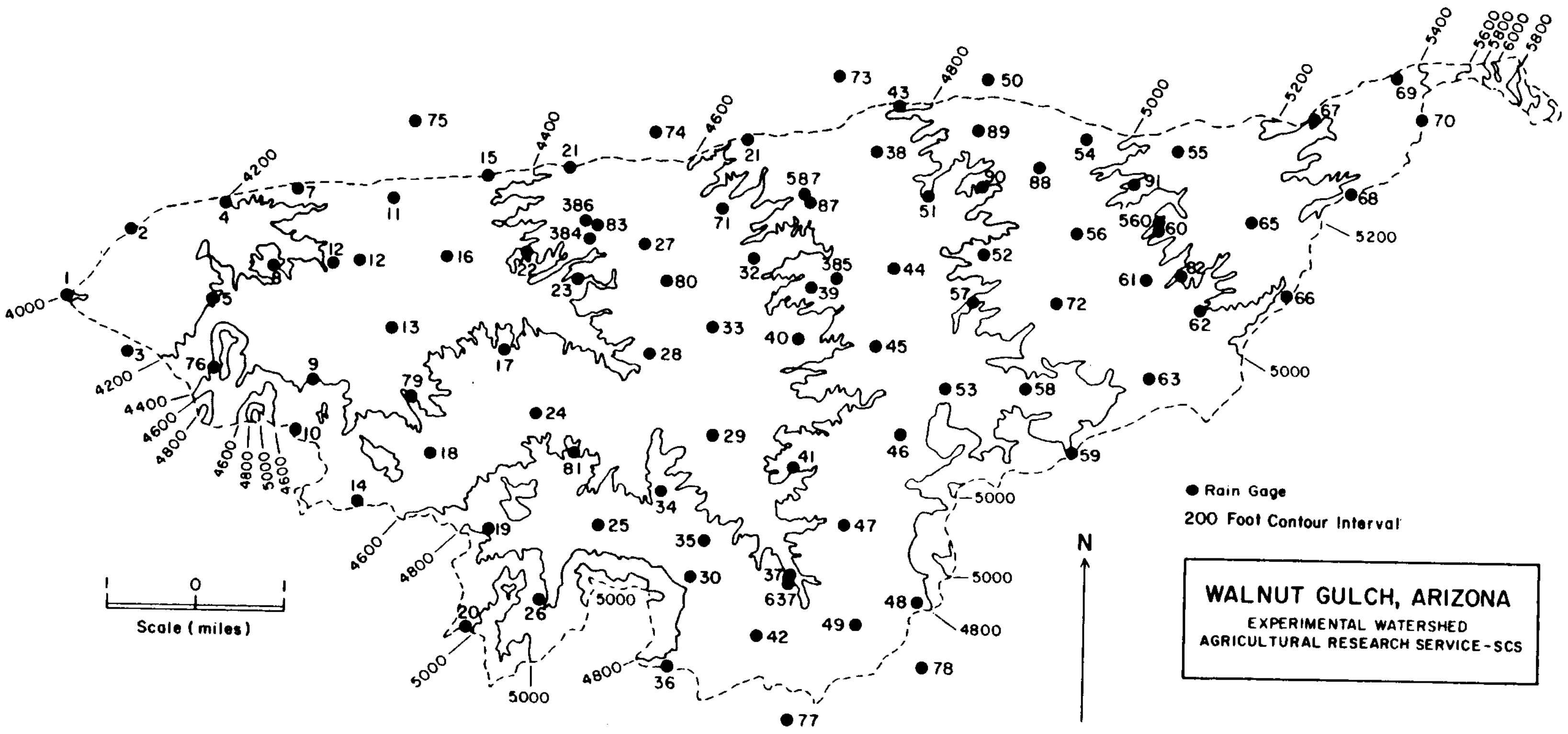

Fig. 3.2.1 Walnut Gulch Experimental Watershed

3.4 The Walnut Gulch Raingage Network

Osborn et al. [1979] report that the Walnut Gulch raingage network was originally designed as a grid with gages located at one mile intervals. Because access to some of the planned locations was difficult due to the terrain, actual gage locations vary from the original grid. Recording raingages were installed and became operational in 1954. Due to a lack of funding, the basic network was incomplete until 1961. On the basis of their research, Osborn and Reynolds [1963] determined that there were "gaps" in the network. Consequently, additional gages were placed within the perimeter of the watershed. By 1967, additional gages had been added along and outside of the watershed boundary. The density of the network is now about one gage per 2 km^2. The total area covered by the network is roughly 180 km^2.

The network is composed of standard non-recording raingages, and 6 and 24 hour weighing type strip chart recording gages. In this study, only data from the recording raingages is used.

CHAPTER 4

Preliminary Data Analysis

4.1 Preliminary Analysis of Long-Term Raingage Statistics

In order to gain a preliminary feeling for the climate in the region of Walnut Gulch, Arizona, an analysis of the long-term record for a single raingage was undertaken. Raingage 29, located roughly in the center of the basin, was selected for this purpose (see Figure 3.2.1).

A continuous record from 1 January 1955 through 31 December 1977 was examined. In order to ascertain the seasonal variation in rainfall, a preliminary analysis of the precipitation data compiled for each month was performed. The results are presented in Figures 4.1.1, 4.1.2, 4.1.3, and 4.1.4 which show respectively the monthly number of showers, the average monthly rainfall, the average duration of the showers and the average depth per shower all at Gage 29. These results clearly show the domination of annual rainfall by the summer events. Almost one half of the annual rainfall events and over one half of the annual rainfall take place during the months of July and August, in showers of duration on the order of 1 to 2 hours.

We define a shower as being a precipitation event during which one or more gages is always recording precipitation.

4.2 Selection of the Summer Rainy Season

Observations by Osborne et al. [1979] show that the summer precipitation at Walnut Gulch stems from air-mass thunderstorms while precipitation during the winter months results from frontal storm systems. Our analysis of the gage 29 records supports these observations. We note

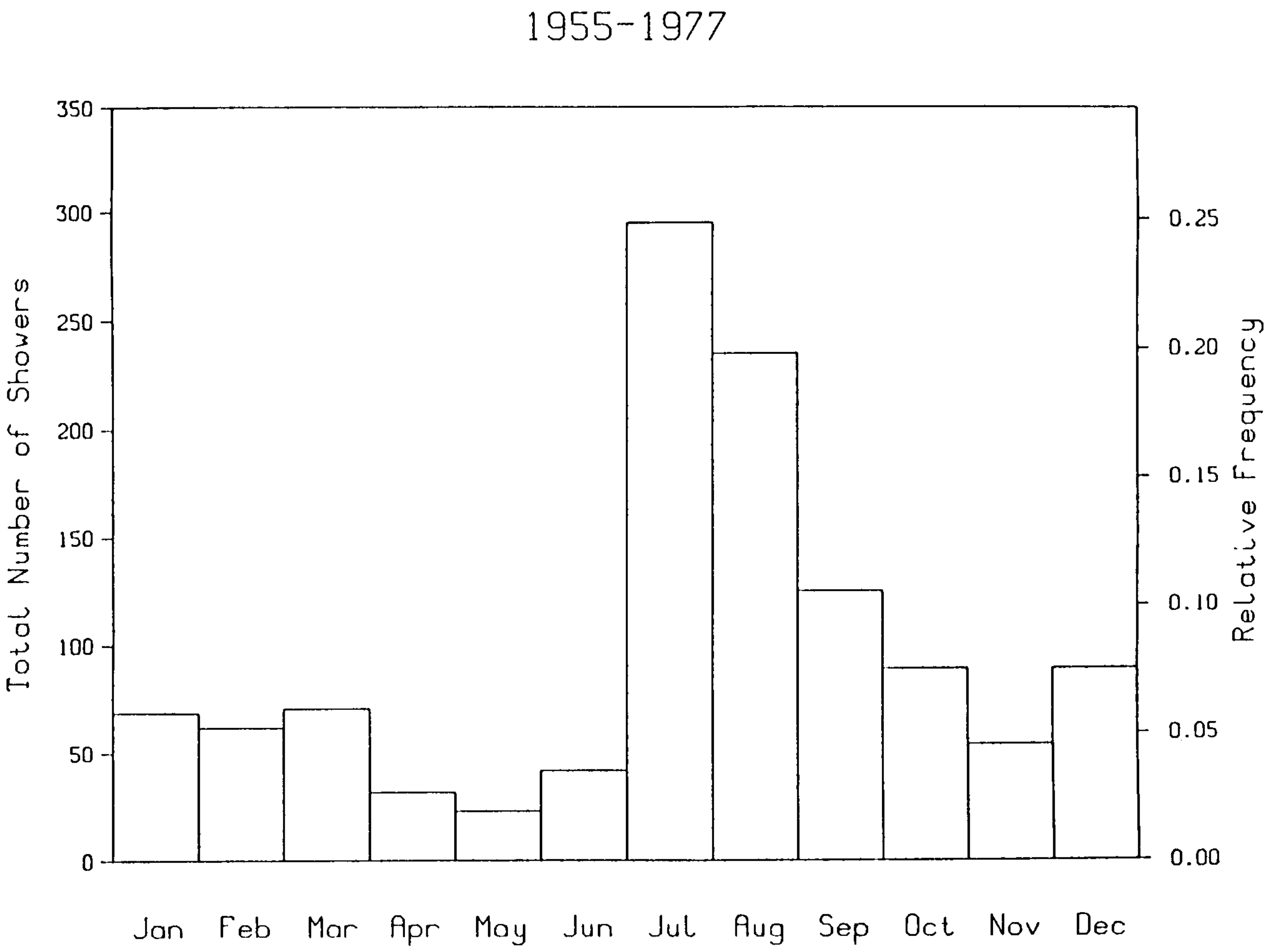

Fig. 4.1.1 Monthly Number of Showers at Gage 29

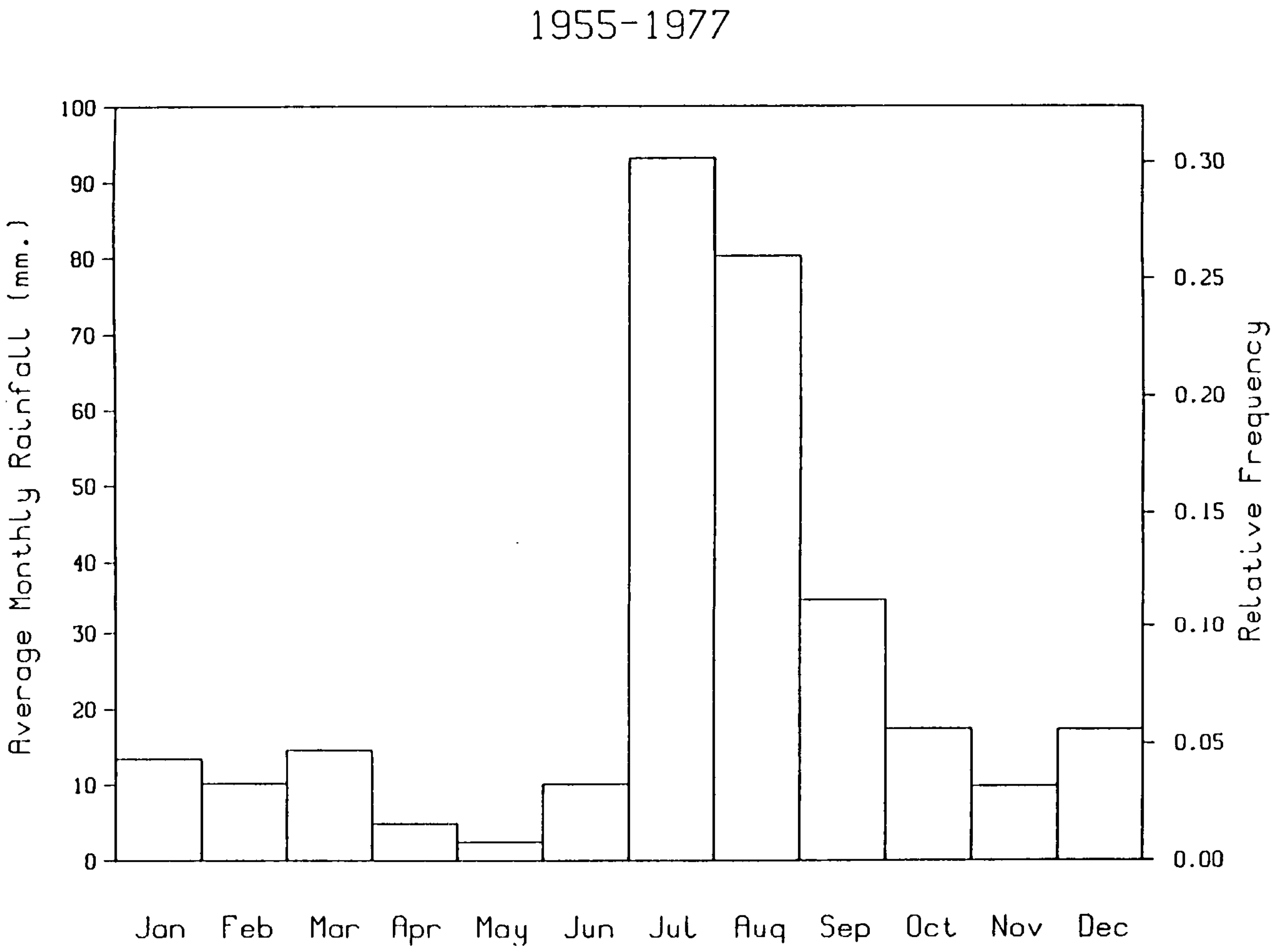

Fig. 4.1.2 Average Monthly Rainfall at Gage 29

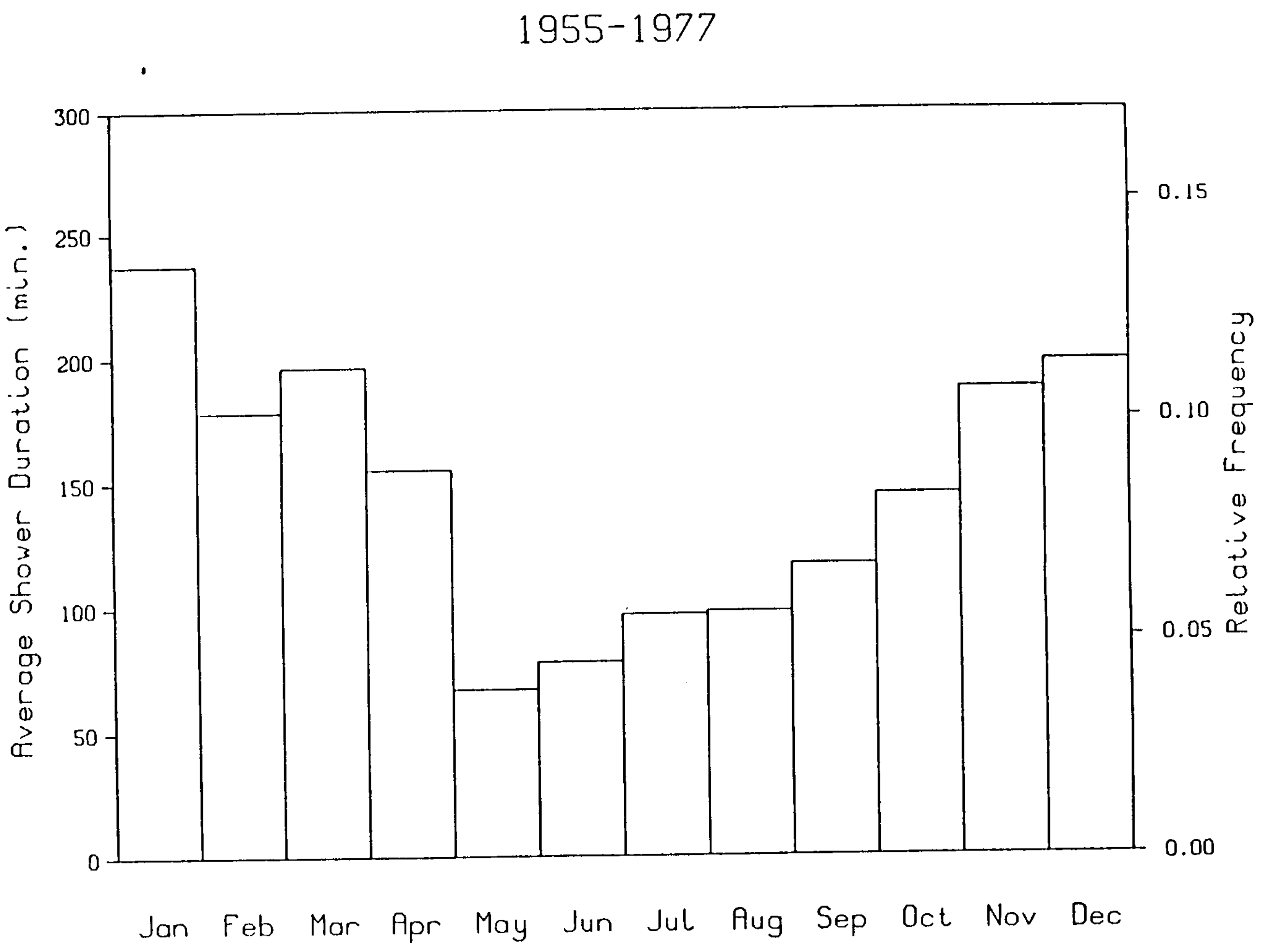

Fig. 4.1.3 Average Shower Duration at Gage 29

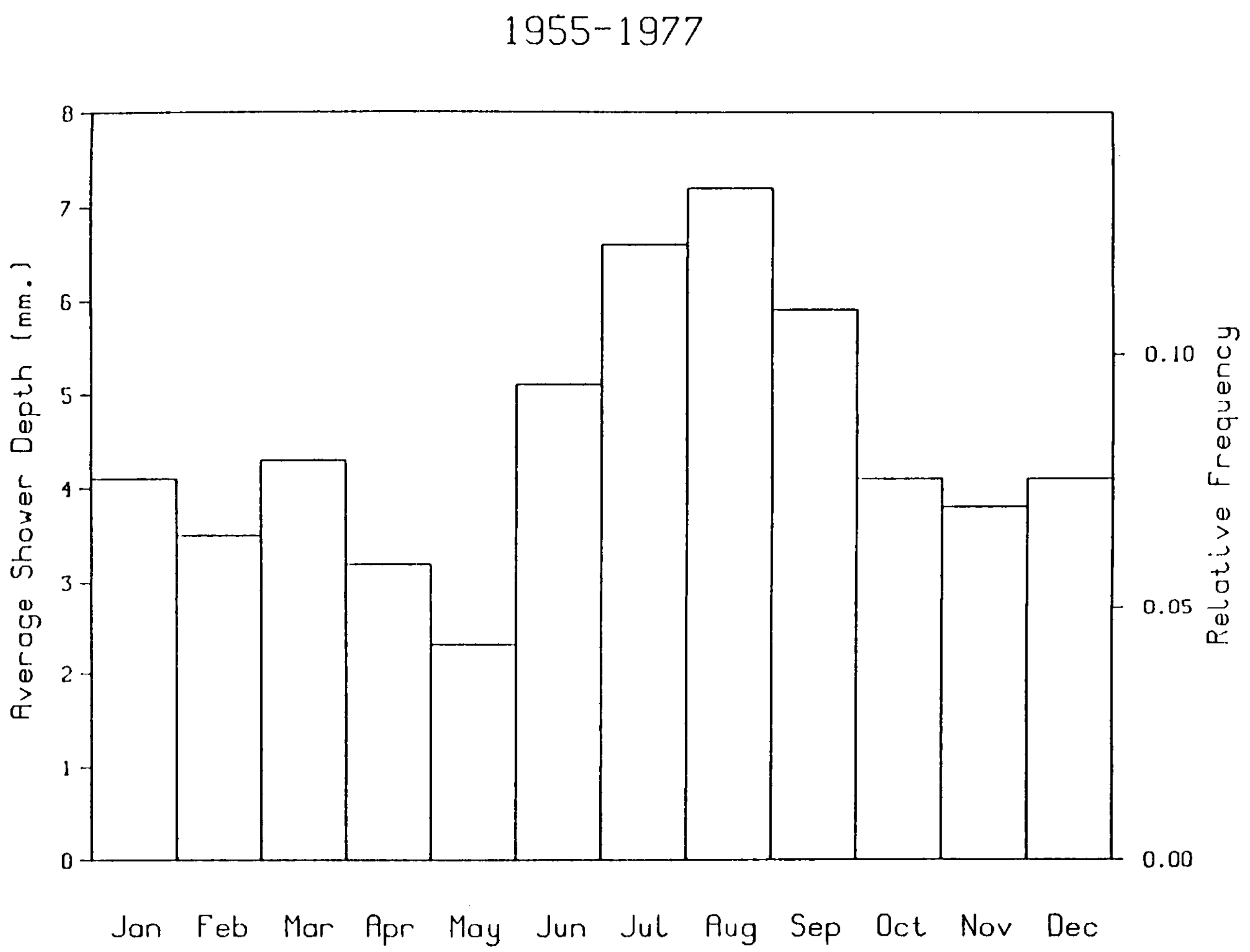

Fig. 4.1.4 Average Depth Per Shower at Gage 29

for example that the average storm duration of the winter storms is twice that of the summer storms.

Because our objective is to analyze the spatial characteristics of precipitation systems having little or no advective component, the summer air-mass thunderstorms are chosen and definition of the summer rainy season is necessary. From the long term record of gage 29, the month of May clearly marks a seasonal transition in the local climate. In May, the total number of showers recorded at this gage is a minimum and the least amount of rainfall was recorded as shown in Figures 4.1.1 and 4.1.2, respectively. It is more difficult to distinguish the month in which the summer season ends. The number of events recorded by gage 29 in September is roughly half that recorded during the month of August, however, the average rainfall depth per shower as shown in Figure 4.1.4 would appear to be more closely related to the summer storms than to the winter storms. For these reasons, the summer rainy season selected for this study was chosen to extend from 1 June through 1 September.

4.3 Independent Rainstorm Events and the Storm Day Concept

The problem of selecting independent events from gage records has no clear cut solution at the present time. Clearly, single raingage observations offer little in the way of information to assist in defining a single storm. Casual observations support this contention. A shower may stop for a brief period of time, only to resume again. Intuitively, if the inter-shower duration is "short", the separate showers probably belong to the same storm system. However, how long should this inter-shower duration be in order to delineate separate storm events? Restrepo-Posada and Eagleson [1982] examined this problem and suggested that should the coefficient of variation of the time between showers be less than or equal to one, then the storms might be assumed statistically independent events. In other words, there is a minimum intershower time in order for two successive showers to be considered statistically independent rainstorms. For Walnut Gulch, analysis of gage 29 summer season records using the method of Restrepo-Posada and Eagleson [1982] indicates this time to be 18.6 hours. A similar analysis which incorporates all the gages in the network simultaneously in time, indicates that for the entire Walnut Gulch watershed, the minimum intershower time is 27.5 hours.

The records for gage 29 suggest that during the summer rainy season, there is precipitation every two to three days, particularly during July and August. In a climate dominated by air-mass thunderstorms, one would expect the highest frequency of showers to occur in the mid- to late afternoon when the land surface temperatures and therefore the vertical convective processes are at a maximum. The gage 29 records are analyzed for shower starting time and shower duration during the summer season. Figure 4.3.1 shows that rainfall begins most often between 4 and 5 p.m.

On the basis of this analysis and the minimum intershower time for independent events, daily precipitation amounts are considered to be independent. Accordingly, a "storm day" is defined as the 24 hour period during which rainfall is recorded by at least one raingage within the basin network. The "storm day" begins at noon of the calendar day and extends to noon the following day. During the eight year period of record for which complete network data were available, there were 431 storm days in the Walnut Gulch basin.

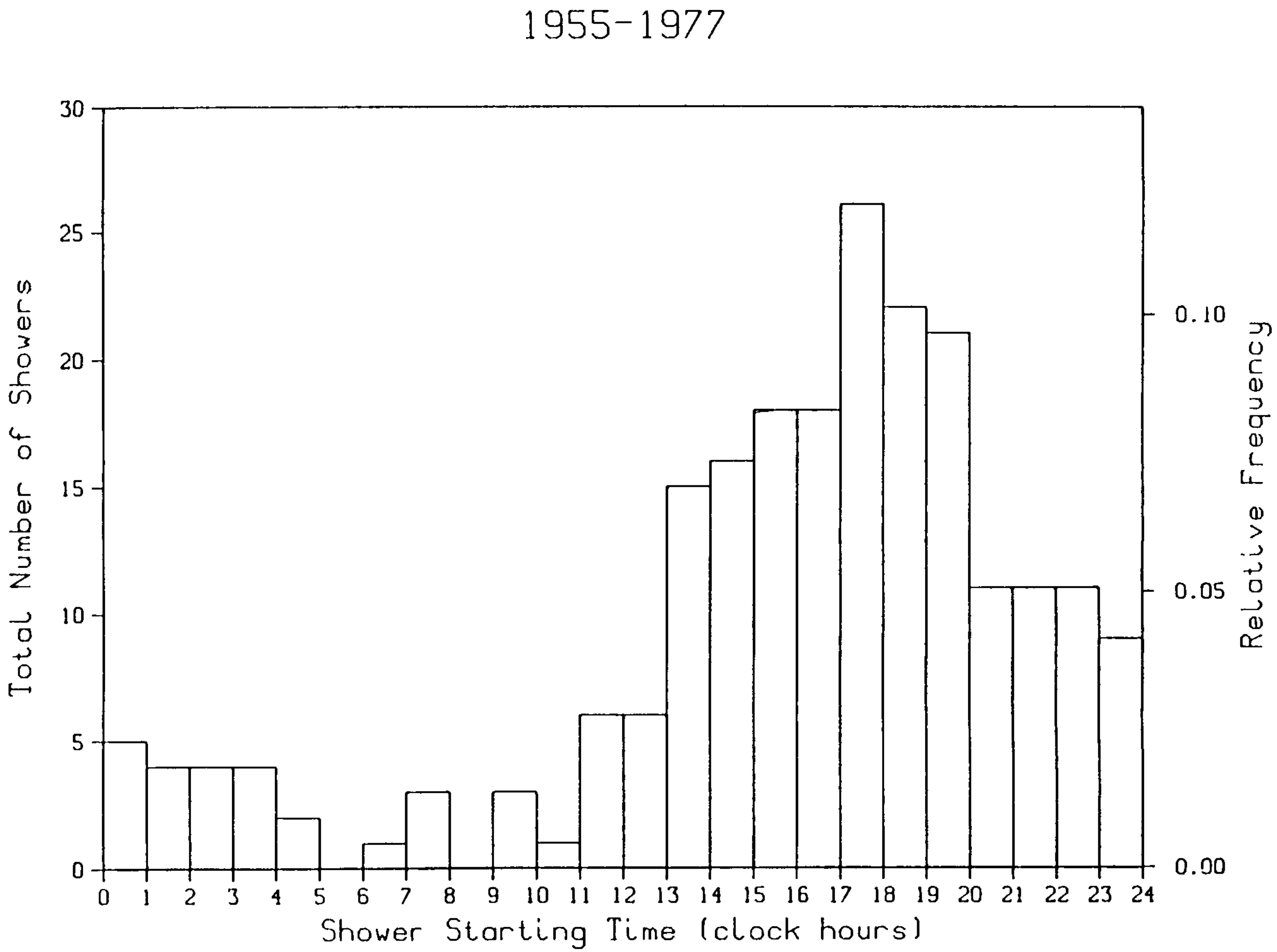

Fig. 4.3.1 Shower Starting Time at Gage 29

A computer program called SUMMER.FORTRAN is written which retains only the summer season station observations. SUMMER.FORTRAN is discussed in Chapter 12.

CHAPTER 5

Data Reduction and Reformulation

5.1 Organization of the Raw Data

The data used for the spatial analysis of storm depth are drawn from eight years of digitized raingage records. Information from over one hundred gages is available and ultimately the data from 93 gages are used. The majority of the raingages are of the tipping bucket variety used with continuously recording strip chart devices which record cumulative rainfall.

According to Agricultural Research Service personnel, rainfall intensity is digitized from the recorder charts when there appears to be a change in the rate of total rainfall accumulation. This leads to non-uniform sampling intervals during which the rainfall intensity may be assumed constant.

For the period of time during which each gage recorded continuous rainfall, a separate record was created for the digitized archive. Each record (for each gage) contains the raingage identification number, the (starting) date of the shower, the time at which the shower began, the duration of the shower and the cumulative or total depth of the shower. In the case where there is a lull in the rainfall (which was not unusual), a separate record was created when the rainfall began again. For this study, the relevant information in each record is the date of the shower, the starting time and duration of the shower, and the total depth recorded by each raingage.

5.2 Conceptual Reorganization of the Raw Data

Following the decision to adopt the "storm day" convention discussed in Section 4.2, special preparation of the original raingage data becomes necessary. To reiterate, the storm day is defined as the 24 hour period beginning at noon of the present calendar day and extending to noon the following day. Consider a hypothetical shower which was recorded by raingage 46 on 27 July 1971. The shower began at 1:00 p.m. and ended at 2:47 p.m. In the case of this record, the total rainfall from this shower belongs to that which fell during the 27 July, 1971 storm day. Suppose instead that the shower began at 7:30 a.m. on 27 July 1971 and ended at 11:02 a.m. the same morning. Then by our definition, the rainfall recorded during this time is assigned to the total rainfall recorded during the 26 July 1971 storm day. Suppose however that the shower began at 11:00 a.m. and ended at 2:00 p.m. on the 27th of July. In this case, the duration of the shower recorded by the gage was 3 hours. We therefore assume a constant rainfall intensity and assign one third of the total depth from this record to the 26 July 1971 storm day, and the remaining two thirds to the 27 July 1971 storm day.

The objective at this stage of data manipulation is to sum the total depth of rainfall recorded by each gage for each storm day according to our definition. There are many instances in the original data in which rainfall was recorded in the form of several separate records which were obtained during the course of a single storm day. In order to accomplish this manipulation, a computer program is developed and called DAY.FORTRAN. This program is discussed in Chapter 12.

A brief example is presented below which illustrates the basic principles of the DAY.FORTRAN computer code. We shall continue with the hypothetical data recorded by gage 46 on 27 July 1971.

The original data are written in the following format.

Table 5.2.1

Original Station Record Data

Gage I.D.	Year	Month	Day	Starting Time	Duration (min)	Total Depth (mm)
46	71	7	26	1300	60	5.6
46	71	7	27	1100	180	9.0
46	71	7	27	1900	120	11.3
46	71	7	28	330	80	6.2
46	71	7	28	1400	75	3.1

We see that the first shower began at 1:00 p.m. and ended at 2:00 p.m. and the total depth from this shower is assigned to the 26 July 1971 storm day. The first shower recorded on the 27th began at 11:00 a.m. and ended at 2:00 p.m. In this case, 3 mm or one third of the total shower depth is assigned to the total storm day depth of 26 July. The remaining 6 mm (that which fell after noon on the 27th) is assigned to the total depth of the 27 July storm day. The second shower on the 27th began at 7:00 p.m. and lasted until 9:00 p.m. This depth of 11.3 mm is assigned to the 27 July storm day total. The first shower which was recorded on the 28th of July began at 3:30 a.m. and ended at 4:50 a.m. the same morning. Because this shower began and ended before noon on the 28th, the total shower depth of 6.2 mm is assigned to the 27 July 1971 storm day. The final shower in this hypothetical series of records began at 4:00 p.m.

and ended at 5:15 p.m. on the 28th. The total depth of this shower is assigned to the 28 July 1971 storm day.

From the five original records in this hypothetical series, the DAY.FORTRAN program creates the following new data set, each record in the new series corresponding to each of the three storm day total depth records.

Table 5.2.2

DAY.FORTRAN Created Station Record Data

Gage I.D.	Year	Month	Storm Day	Total Depth (mm)
46	71	7	26	8.6
46	71	7	27	23.5
46	71	7	28	3.1

Following the creation of the new data set for each of the raingages during the eight summer seasons of records, a preliminary assessment of the gage records was conducted. Of the one hundred plus gages among the records, several were not operational for the entire period of record, and so they were discarded from further processing. Several others were not located on the map provided by the U.S.D.A. and were also discarded from further processing. Ultimately, 93 raingage records were included in all further aspects of the study. There are eight years of record during which all 93 raingages were operating. These yielded 431 separate storm days.

5.3 Basin-wide Inter-gage Comparison and Data Creation

The next phase of data manipulation involves the inter-gage comparison of all the raingages throughout the network for each storm day. As was described in the preceding section, the original data records for each shower include only periods during which rainfall was recorded.

The next step is to create storm day records which note those gages that recorded no rainfall during each storm day of a single summer season. In order to accomplish this task, a computer program called BALANCE.FORTRAN was developed and run for each of the eight summer rainy seasons. The BALANCE.FORTRAN computer code is discussed in Chapter 12.

Following the summer season by summer season implementation of DAY.FORTRAN, BALANCE.FORTRAN is run. BALANCE.FORTRAN first inspects the storm day records for the season gage by gage and determines the number of storm days for the season, and the precise date of each storm day. The program then reinspects each gage storm day record to see if that gage recorded precipitation during a particular storm day. If that gage did record rainfall, the program moves on to the next gage record. If, however, that gage did not record rainfall on that storm day, then a record of zero rainfall depth is created. This comparison and creation of zero depth rainfall records is conducted on each gage as necessary for each storm day, summer season by summer season, for each of the eight years of record.

The following is an example which will clarify the approach of the BALANCE.FORTRAN program. Consider the results of the DAY.FORTRAN processing outlined in the preceeding section for gage 46. Here we also present the hypothetical results of the DAY.FORTRAN processing of gage 15. This data is presented in Table 5.3.1. For the sake of brevity, we assume that the 1971 summer rainy season had only five storm days.

Table 5.3.1

DAY.FORTRAN Created Station Data Record

Ggge			Storm	Total
I.D.	Year	Month	Day	Depth (mm)
46	71	7	26	8.6
46	71	7	27	23.5
46	71	7	28	3.1
15	71	6	24	5.9
15	71	7	27	20.2
15	71	7	28	2.1
15	71	8	1	4.3

From the table above, we see that there were storm days on 24 June, 26 July, 27 July, 28 July, and 1 August. Following implementation of BALANCE.FORTRAN, zero total depth records are created for the storm days of 24 June and 1 August for gage 46, and for 26 July at gage 15. The final data set is presented in Table 5.3.2.

Table 5.3.2

BALANCE.FORTRAN Created Storm Day Record

Gage			Storm	Total
I.D.	Year	Month	Day	Depth (mm)
46	71	6	24	0.0
46	71	7	26	8.6
46	71	7	27	23.5
46	71	7	28	3.1
46	71	8	1	0.0
15	71	6	24	5.9
15	71	7	26	0.0
15	71	7	27	20.2
15	71	7	28	2.1
15	71	8	1	4.3

CHAPTER 6

Interpolation and Filtering of the Storm Day Random Field

6.1 The Numerical Watershed Boundary

The random field of rainstorm depth must be analyzed over a particular area. It is convenient and hydrologically relevent to define the area by the boundaries of the Walnut Gulch catchment.

For computational purposes, it is convenient to interpolate the 93 station observations onto a rectangular mesh of equally spaced node points. The resolution of the mesh is dictated by the available virtual memory of the computer system. The finest grid resolution (node spacing) in this study is 100 meters. This produces 41,160 equispaced node points, or a matrix of 140 rows by 294 columns. A scale map of the U.S.G.S. topographic series with the raingage locations and the watershed boundary was obtained from the U.S.D.A. A rectangular mesh of lines representing the row and columns are ruled on this map at 100 meter spacings (see Fig. 6.1.1). Each 100 meter by 100 meter box represents a 0.01 km^2 grid square. The intersection of each row and column represents a node point. Horizontal and vertical line segments approximating the perimeter of the drainage basin are drawn on the map between nodes lying close to or on the basin boundary.

Following the creation of this mesh boundary which approximates the actual catchment boundary, the grid is examined row by row for the coordinates of each node which lie on the approximate boundary. For each row, a special logical examination is made and a unique expression is created. The purpose of this expression for each row is to replace the values of the interpolated point depths outside the watershed boundary

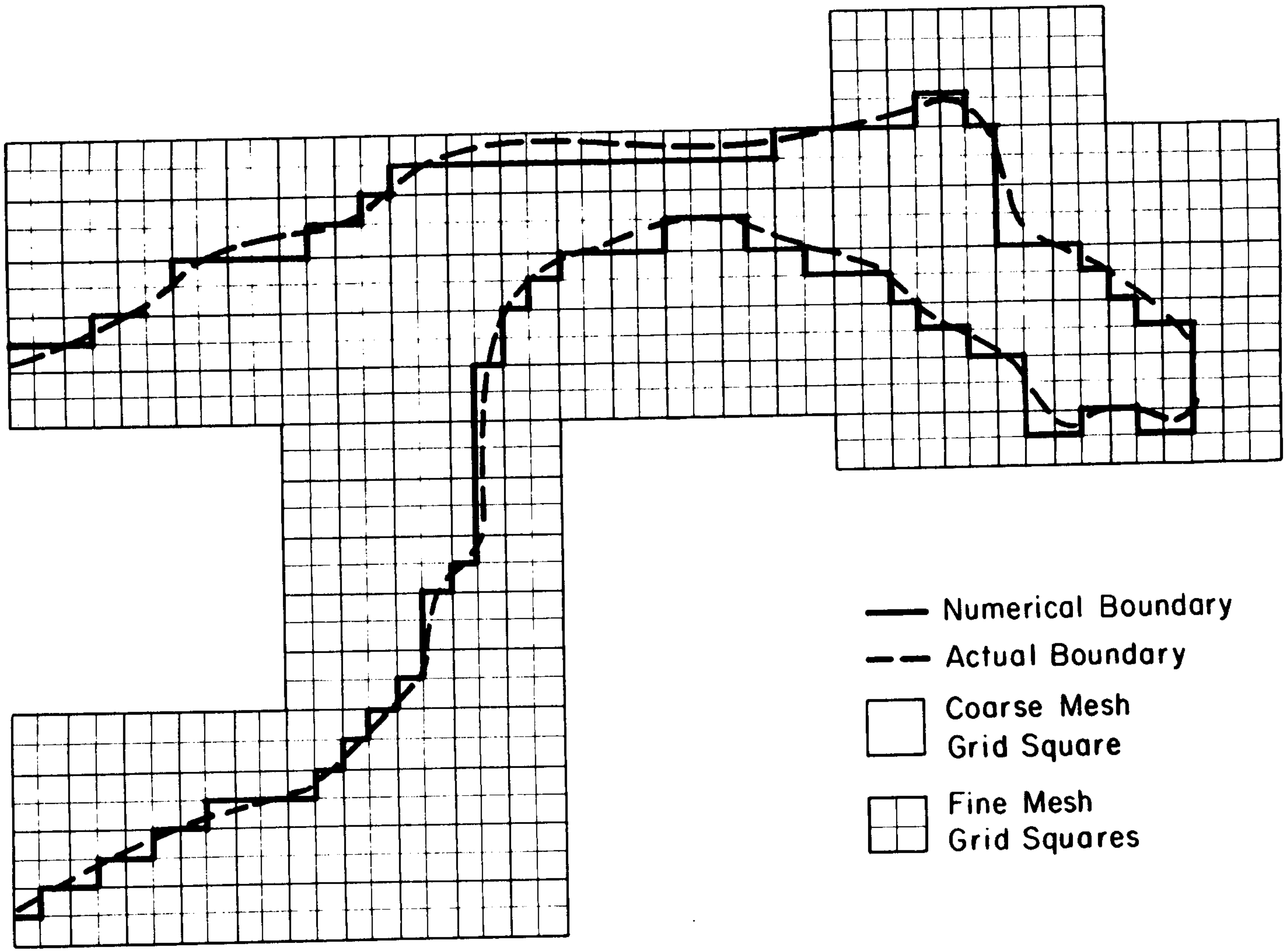

Fig. 6.1.1 The Grid Mesh and Numerical Watershed Boundary

with an artificial value. For example, on row 35 there are two artificial boundary nodes, i.e. row 35 crossed the basin boundary twice. The first boundary node lies in column 86 and the second in column 184. The appropriate logical expression for row 35 is then that all values of the interpolated mesh surface on row 35 that lie in columns 86 through column 184, are within the limits of the basin. For columns 1 through 85 and columns 185 through column 294 the values on row 35 lie outside the area of interest and we wish to ignore them. For each of these latter values regardless of the value assigned to those nodes by a surface interpolator, each node is assigned a value of -1.0 mm which defines nodes of no interest. The assignment of this nonsense value of storm day depth then becomes the key to the various spatial sampling schemes.

From the scale map, the coordinates of each raingage is determined. The coordinate system is based solely on the coordinates of the ruled mesh. A special computer program is developed that assigns coordinates to each raingage in each storm day record. The program is called COORD.FORTRAN and is discussed in Chapter 12. Thus, given the total storm day rainfall depth and spatial coordinates for each of the 93 gages in the final network, the bivariate surface interpolator can be implemented.

It should be noted that while the natural catchment area as given by the A.R.S. is 150 km^2 the area as defined by this numerical boundary is 154 km^2 and will be used hereafter.

6.2 The Bivariate Surface Fitting Interpolator

In Chapter 5, we discussed the basin wide inter-gage comparison and data creation which resulted in 431 storm days for the eight year summer season period of record. At this stage of development, there are 93 raingage records for each storm day ready for further processing. In order to proceed with the planned spatial analyses, an algorithm which interpolates a continuous surface from the gages is necessary.

For each storm day gage record, we have created data which include total storm day depth and its x and y spatial coordinates in the 2-d plane. A particular complexity is that the Walnut Gulch raingage network is not evenly spaced on a grid. The surface interpolation algorithm developed by Akima [1978] is adopted to interpolate a bivariate surface from irregularly distributed data points.

In Akima's [1978] procedure, the x-y plane is divided into a number of triangular cells, each having projections of three data points in the plane as their vertices. A bivariate fifth degree polynomial in x and y is developed for each cell. Estimated values of the partial derivatives at each vertex of the cell are used to determine the coefficients of the polynomial.

The interpolation of depth Y(x,y) within each triangle is based on the following:

(1) The value of the function at point (x,y) in a triangular cell is interpolated by a bivariate fifth-degree polynomial in x and y; i.e.

$$Y(x,y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} x^j y^k \tag{6.2.1}$$

(2) The values of the function and the estimates of its first and second partial derivatives are given at each vertex of the cell.

(3) The partial derivatives of the function taken in the direction perpendicular (or normal) to each side of the cell are at most third-degree polynomials in the variable measured in the direction of the side of the triangle.

Akima [1978] also proved that the interpolated surface must be smooth due to continuity of the derivatives along the sides of adjacent triangluar cells. He did not expect a high degree of accuracy with respect to extrapolation outside the limits of the spatial data, but felt that it would be desirable for any extrapolations to be smooth.

The bivariate surface interpolator developed by Akima and used for this study is proprietary software. Access to the source code in order to more fully describe the techniques used is not possible. The MIT computational group subscribes to the I.M.S.L. (International Mathematical and Statistical Libraries). The subroutine developed by Akima is called IQHSCV by I.M.S.L.

Given the raingage storm day total depths and their spatial coordinates in the x-y plane, the bivariate surface fitting interpolator outputs a grid of uniformly spaced interpolated data at the nodes of the mesh. The outside dimensions of the mesh in real space are 14.0 km. by 29.4 km., a rectangular area which encompasses the entire 150 km^2 Walnut Gulch drainage basin plus those gages located outside its perimeter.

Two mesh resolutions were used for the various probabilistic spatial analyses in this study. For one part of the work, a mesh resolution of 200 meters (0.04 km^2 grid square area) was used and for the second portion, a node spacing of 100 meters (0.01 km^2 grid squares) was used. More details about each will be discussed in Chapter 7 and Chapter 8.

It is realized that use of a deterministic interpolator will introduce spatial dependence that is not present in the observations. This will bias estimates of the moments and of the spatial correlation to a degree that will be evaluated empirically in a later section.

6.3 Additional Post Surface Interpolation Filters

The interpolated surface outside the watershed boundary is overlaid by the special boundary filter which imposes values of -1.0 mm of total storm day depth at the node points outside the basin boundary. In addition to this filter, two other filters are imposed on the random field.

In the mathematical modeling of spatial distribution of total storm depth, presented in a companion volume, an important assumption is made about the spatial distribution of storm depth in a given raincell. It is assumed that the minimum non-zero depth is 0.01 mm. This is consistent with there being a limit to the practical resolution of rainfall recording devices. Therefore, a filter is created that examines the mesh and imposes a value of zero on all those random field nodes which have total storm day depths less than or equal to 0.01 mm. This includes those nodes at which, due to the smoothing and continuous nature of the bivariate surface interpolator, there are node point depth values less than zero. The filter is designed in such a way as to redefine only those node point depth values within the perimeter of the watershed and the previously defined values of -1.0 mm. outside the basin are left untouched.

Following the spatial sampling of the random field within the limits of the watershed, these values are redefined by another filter. The purpose of this third filter is only for creating contour plots. In such cases any node point with a value of zero, is redefined to have a value of -0.99 in order to allow the graphics package to draw a zero rainfall isohyet. At no time are the results from this last filter used during any of the spatial sampling schemes.

CHAPTER 7

Storm Day Observations: Sampling from the Coarse Grid Mesh Random Field

7.1 Spatial Homogeneity and Isotropy of the Random Field

In the sampling process, some basic assumptions about the nature of the random field are necessary. In particular, spatial homogeneity and isotropy play a crucial role in the theory and application of random fields. The random field, $Y(\underline{x})$, is "wide sense" homogeneous if (Bras and Rodriguez-Iturbe [1985]).

$$E(Y) = \text{constant} \qquad (7.1.1)$$

and

$$\text{COV}(\underline{x}_1,\underline{x}_2) = \text{COV}(\underline{x}_1-\underline{x}_2) = \text{COV}(\underline{\nu}) \qquad (7.1.2)$$

or

$$\rho(\underline{x}_1,\underline{x}_2) = \rho(\underline{x}_1-\underline{x}_2) \qquad (7.1.3)$$

$$= \rho\ (\underline{\nu})$$

The random field $Y(\underline{x})$ is considered to be homogeneous and isotropic if

$$\rho(\underline{\nu}) = \rho(|\underline{\nu}|) \qquad (7.1.4)$$

This implies that the correlation function depends only on the length of the vector $\underline{\nu}$, and is independent of its direction in space.

In the ideal case of a homogeneous random field, the expected value of total rainstorm depth in space can be described as a plane parallel to the surface of the 2-d x-y plane. In practice however, one would expect, some local variation about the mean due to local geographical and topographical influences. Such variations could manifest themselves as local rainfall maxima and minima due to orographic effects, bodies of water, etc. This is investigated here by analyzing the 1970-1973 summer season ensemble of 202 storm days. The expected value and standard deviation of the average interpolated storm day rainfall depth are determined and presented as isopleths over the catchment in Figures 7.1.1 and 7.1.2. The spatial distribution of the average and standard deviation of the storm day depth are also presented in Figures 7.1.3 and 7.1.4 respectively. Inspection of these results suggest that since the surfaces of the mean and variance of the storm day depth are so uniform, the assumption of homogeneity is valid.

The assumption of anisotropy is investigated by comparing average spatial correlations in orthogonal directions. The storm day spatial correlation functions are averaged across the ensemble of storm days in the 1970-1973 period at each lag (up to 9 km) in the x (i.e. West to East) direction and again in the y (South to North direction). These are compared with the bidirectional average in Figure 7.1.5. We see that the three correlation functions are essentially identical and hence the assumption of isotropy seems justified and is used in all further analysis. The techniques used to sample the random field for the spatial correlation are described in Section 7.3.

The apparent non-zero asymptote of ( ) in Figure 7.1.5 represents the correlation arising from the component of variance between storms [Rodriguez-Iturbe, Cox and Eagleson, 1986]. This interstorm variance results from the different strengths of separate storm-producing instabilities. It may also result from incomplete (and variable) coverage of the catchment by the storm [see Eagleson and Wang, 1985]. In our data analysis we have assumed complete coverage.

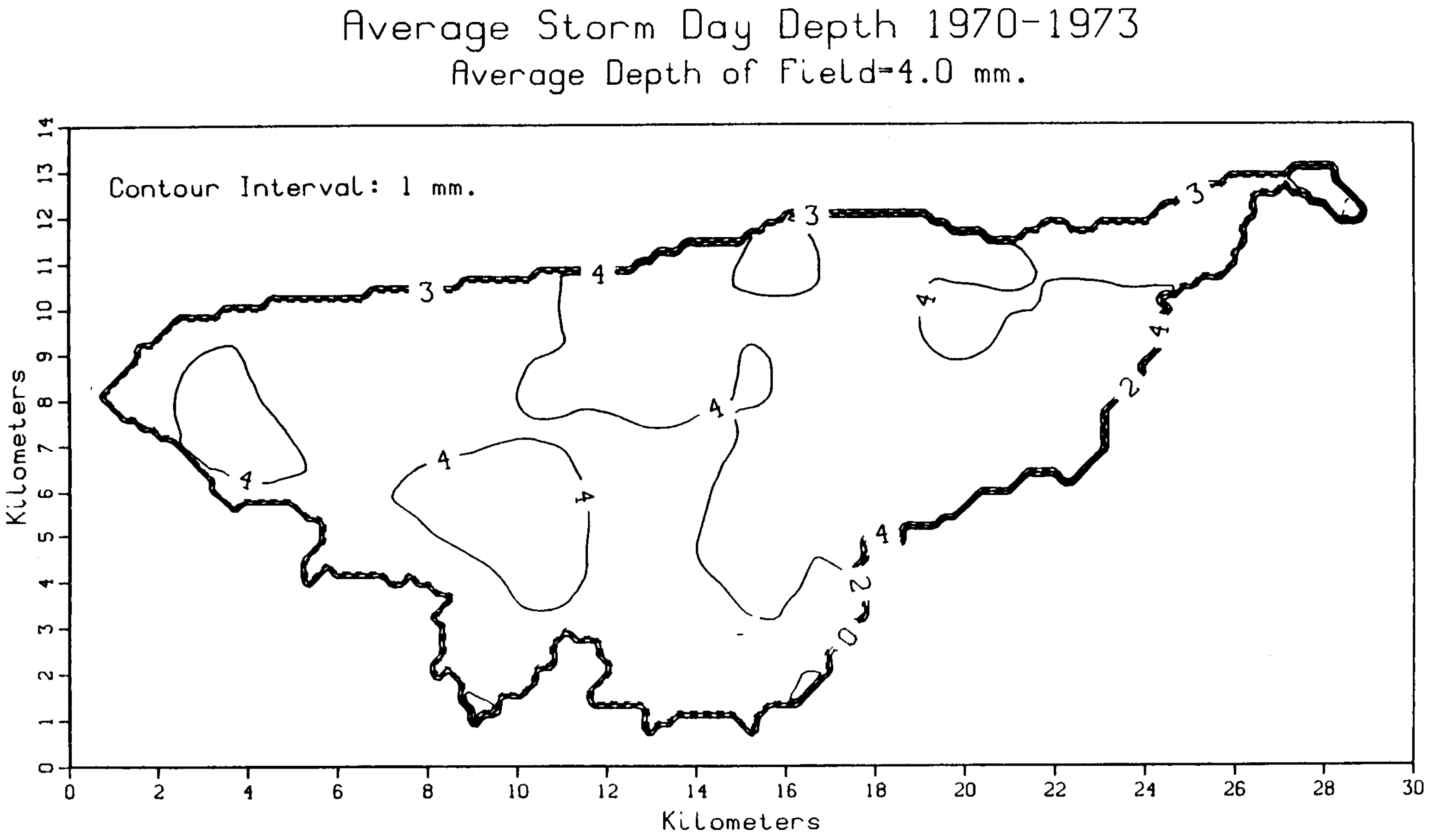

Fig. 7.1.1 Average Interpolated Storm Day Depth

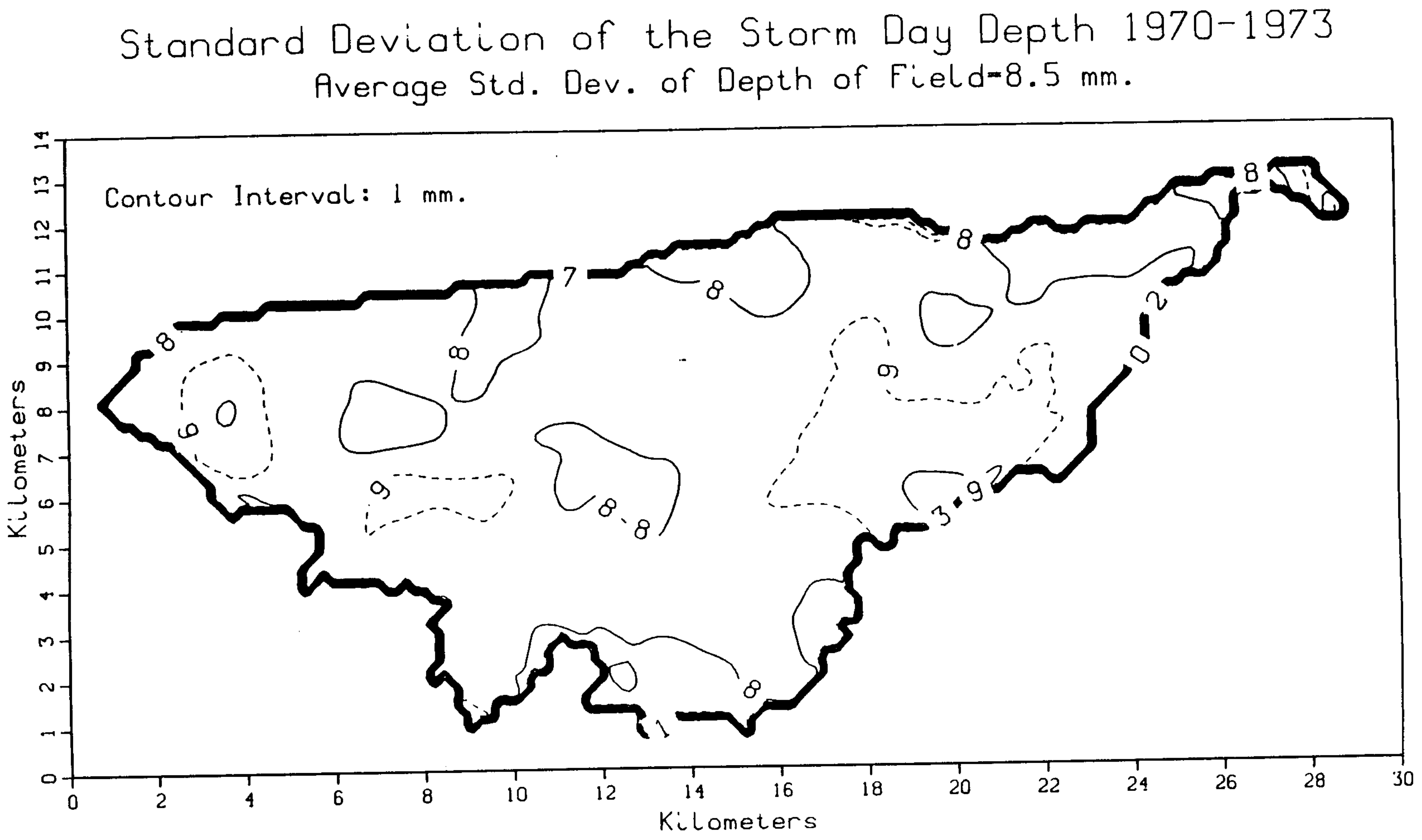

Fig. 7.1.2 Standard Deviation of the Interpolated Storm Day Depth

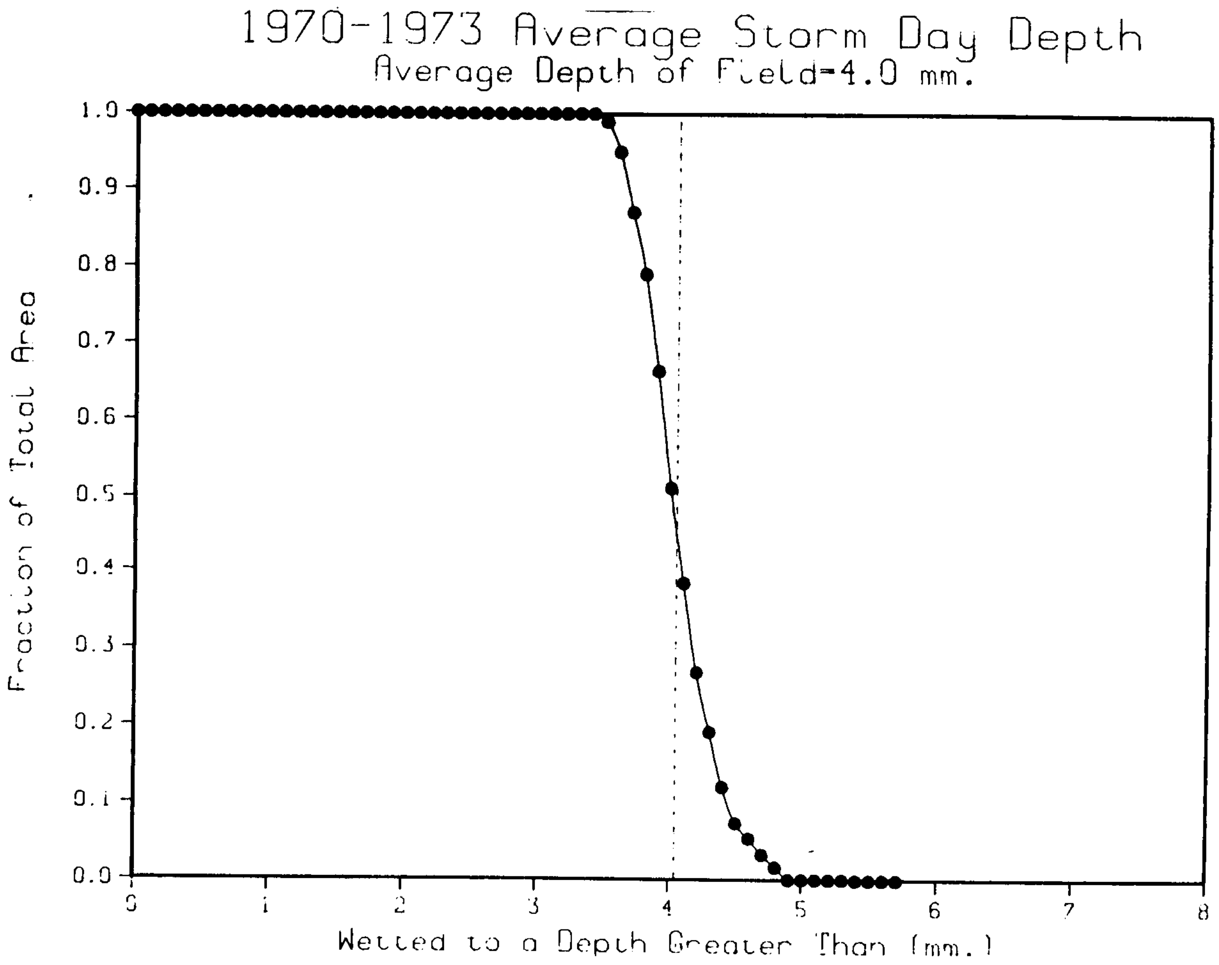

Fig. 7.1.3 Spatial Distribution of the Average Interpolated Storm Day Depth

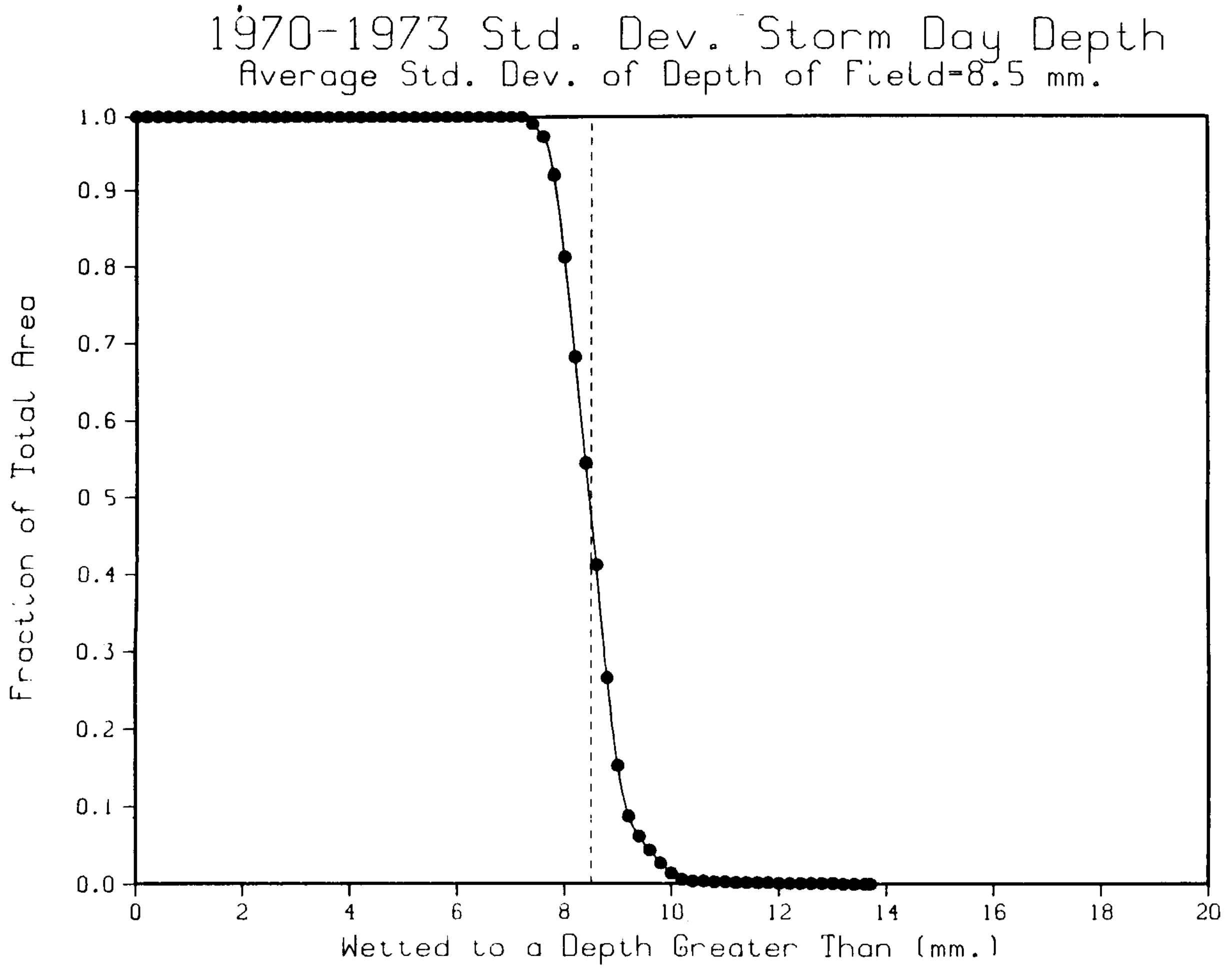

Fig. 7.1.4 Spatial Distribution of the Standard Deviation of the Interpolated Storm Day Depth

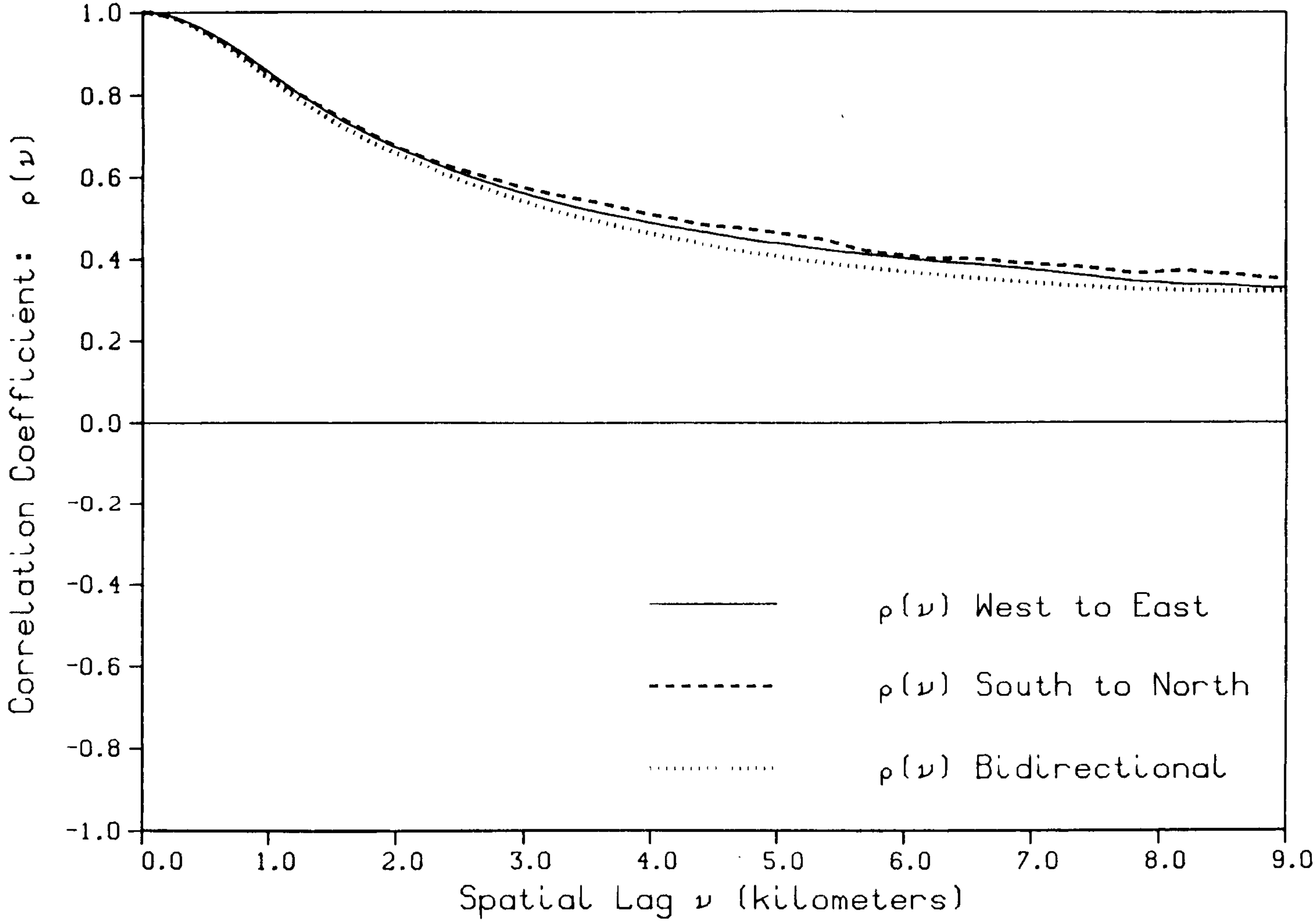

Fig. 7.1.5 Average Storm Day Spatial Correlation

7.2 Sampling for the Moments of Total Storm Day Depth

The first step in the spatial analysis of total storm day rainfall depth Y is to determine the moments of the random field. We will find E(Y), VAR(Y) and C.S.(Y): the expected value; the variance; and the coefficient of skewness respectively. The first two moments are used in estimation of the parameters of the three models of storm depth presented in a companion volume.

Following the creation of the coarse mesh total depth surface by interpolating the 93 station observations and implementing the basin boundary filter described in Chapter 6, the random field is sampled. By invoking a simple logical condition filter, only the random field within the basin boundary is sampled. The moments of the random field for each storm day are estimated in accordance with Benjamin and Cornell [1970].

Let y = y(i,j) represent the total depth at node i,j within the basin limits. Provided that y does not equal -1.0 mm, the estimator of the expected value of point depth E(Y) within the IxJ matrix (i.e. within the catchment) is described by

$$E(Y) = \frac{1}{n} \sum_{i=1}^{I} \sum_{j=1}^{J} y(i,j) \qquad (7.2.1)$$

where n equals the number of node points which lie within the watershed. The estimator of the variance of the point depth within the catchment is given by

$$VAR(Y) = \frac{1}{n-1} \sum_{i=1}^{I} \sum_{j=1}^{J} y(i,j)^2 - E^2[Y] \qquad (7.2.2)$$

and the skewness coefficient of the point depth is given by

$$C.S.[Y] = \frac{\frac{1}{n} \sum_{i=1}^{I} \sum_{j=1}^{J} [y(i,j) - E[Y]]^3}{(VAR[Y])^{3/2}} \qquad (7.2.3)$$

The value of the constant "n" in each estimator equals 4005 which is the number of nodes of the coarse mesh within the catchment boundaries.

It should be noted that the above estimators for the mean and variance of the first two catchment moments are "unbiased". However, because the catchment size is small with respect to the scale of fluctuation of the random fields, most statistical properties of the random field contained within the catchment are small samples of the assumed homogeneous and hence infinite parent random field. Vanmarcke [1985, p. 332] evaluates the bias in such small sample estimators of the parent field statistics. We will deal with this bias in a later section.

The effect of the interpolator on the moments can be seen in Table 7.2.1 where they are compared with the moments of the observations for the two representative storms illustrated in Fig. 7.2.1.

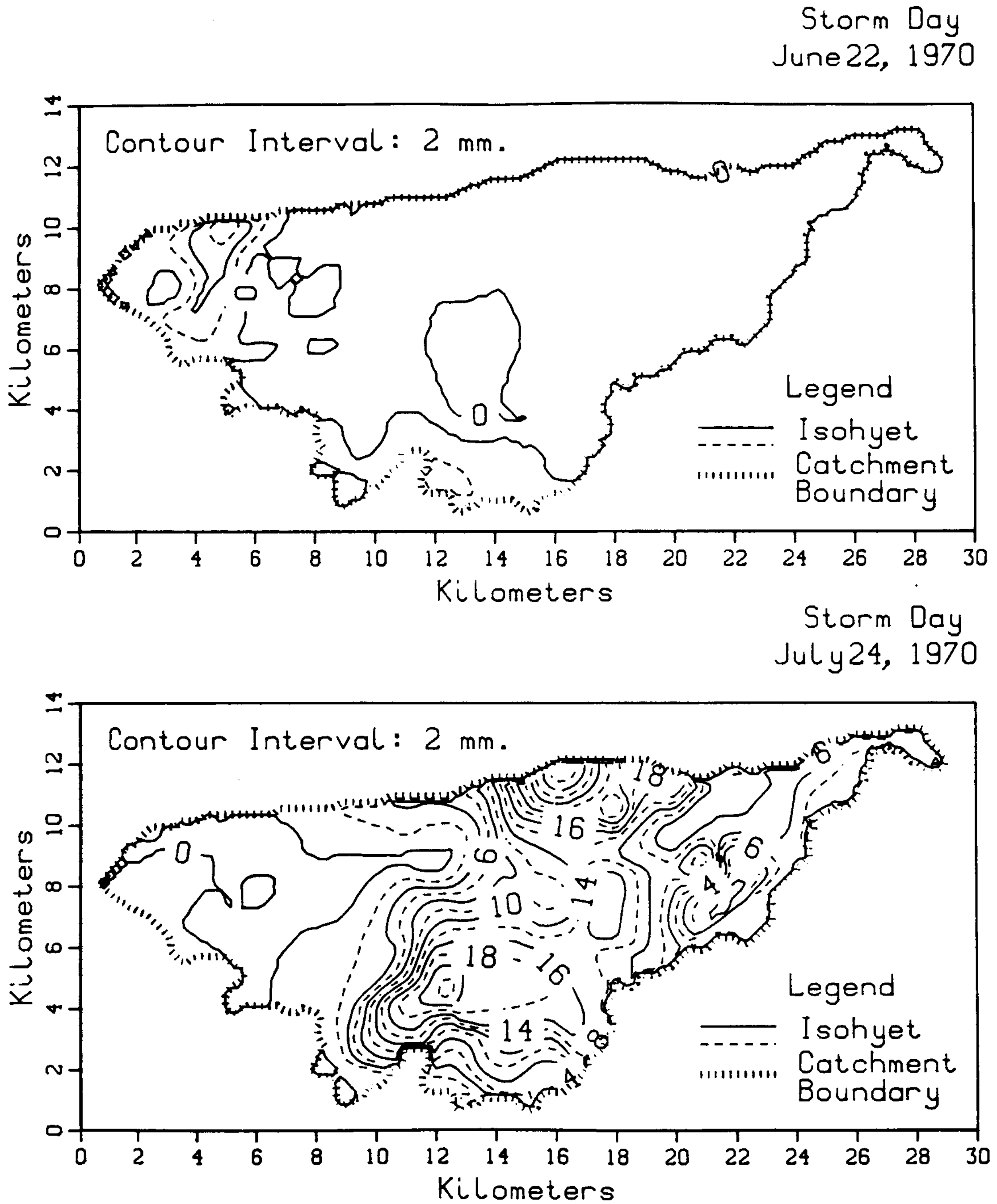

Fig. 7.2.1 22 June 1970 and 24 July 1970 Storm Days

Table 7.2.1

Effect of the Interpolator on
the Moments of The Random Field

Storm Day	Gage Observations, Y_o		Interpolated Field, Y	
	Mean	Variance	Mean	Variance
22 June 1970	0.24	1.04	0.27	0.77
24 July 1970	8.07	61.78	7.97	48.83

Interpolation leaves the mean essentially unchanged, but, as might be expected, its smoothing significantly reduces the variance. Some such smoothing must be present naturally however.

7.3 Sampling for the Spatial Correlation

In Section 7.1, evidence to support the homogeneity and isotropy of the total depth random field is presented. These assumptions are particularly important in determining the spatial correlation of the point rainfall process.

The variation of an n-dimensional random field $Y(\underline{x})$ at two locations $\underline{x}' = (\underline{x}_1',\ldots,\underline{x}_2')$ and $\underline{x}'' = (\underline{x}_1'',\ldots,\underline{x}_2'')$ is characterized by the correlation function [Vanmarcke, 1983]

$$\rho(\underline{x}_1',\underline{x}_2'') \equiv \rho_Y(\underline{x}_1',\underline{x}_2'') = \frac{COV[Y(\underline{x}'),Y(\underline{x}'')]}{\sigma_Y(\underline{x}')\sigma_Y(\underline{x}'')} \qquad (7.3.1)$$

where

$$COV[Y(\underline{x}'),Y(\underline{x}'')] = E[Y(\underline{x}')Y(\underline{x}'')] - m(\underline{x}')m(\underline{x}'') \qquad (7.3.2)$$

$m(\underline{x}')$ and $m(\underline{x}'')$ are the means and $\sigma_Y^2(\underline{x}') \equiv \sigma^2(\underline{x}')$ and $\sigma_Y^2(\underline{x}'') \equiv \sigma^2(\underline{x}'')$ the variances of $Y(\underline{x}')$ and $Y(\underline{x}'')$ respectively.

From the prior assumption of homogeneity, the covariance function will depend only on the relative position of $\underline{x}'$ and $\underline{x}''$. We shall define

$$\underline{\nu} = \underline{x}' - \underline{x}'' \qquad (7.3.3)$$

as the lag vector whose components are

$$\nu_k = x_k' - x_k'' \qquad k = 1,\ldots,n \qquad (7.3.4)$$

For a two dimensional random field, the covariance function can be represented by two functions, each defined for positive lag distances only [Vanmarcke, 1983, pp. 79]. They are:

$$\mathrm{COV}(\nu_1,\nu_2) = \mathrm{COV}(-\nu_1,-\nu_2) \qquad \nu_1,\nu_2 \geq 0 \qquad (7.3.5)$$

and

$$\mathrm{COV}(-\nu_1,\nu_2) = \mathrm{COV}(\nu_1,-\nu_2) \qquad \nu_1,\nu_2 \geq 0 \qquad (7.3.6)$$

An examination of the spatial correlation is necessary for the purpose of model parameter estimation, which will be discussed in detail in a companion volume.

The actual sampling of the random field and the techniques used to perform the spatial correlation analysis received a great deal of attention during the course of this study. The first approach taken was to examine the spatial correlation of total depth among the 93 station observations alone.

7.3.1 Correlation of Observed Station Depths

The raingages are spaced about one kilometer apart on the average. A few gages are spaced less than 100 meters apart, and the greatest distance between a pair of gages among the 93 used, was 23 km. The average intergage distance is 7.9 km. (i.e. averaging all independent pairs).

A computer program is developed called GAGCORR.FORTRAN in order to examine the spatial correlation of observed raingage storm depth. The algorithm designed to perform this analysis first examines each pair of raingages in the network and the distance separating each member of the pair. Each pair is assigned a unique identification value. This I.D. assures that each pair will be examined only once in the contribution to the covariance estimation at that lag distance. The lag distance between a pair of gages is rounded up or rounded down to the nearest 500 meters.

Let $E(Y_0)$ and $VAR(Y_0)$ be the mean and variance of the storm day depth for the 93 station observations. Furthermore, let

$$\nu = \left[(x_i - x_j)^2 + (y_i - y_j)^2\right]^{1/2} \qquad (7.3.7)$$

represent the lag distance between gage i and gage j, where x_i, y_i and x_j, y_j are the spatial coordinates of gage i and gage j respectively (not to be confused with depth y_{0i} and y_{0j} at gages i and j).

Then the spatial correlation at lag k is estimated by:

$$\rho(\nu_k) = \frac{\mathrm{COV}[Y_{o_i}, Y_{o_j}]}{\mathrm{VAR}(Y_o)} \qquad (7.3.8)$$

where $k = 0, 0.1, 0.2, \ldots, 10$ km.

Typical gage correlations are shown in Figures 7.3.1 and 7.3.2 using the 22 June 1970 and the 24 July 1970 storm days (see Figure 7.2.1) as examples. For sparse storms such as 22 June 1970 the results are extremely noisy. Because of this, the analysis of the spatial correlation of rainfall depth is approached using the interpolated field.

7.3.2 Correlation of the Interpolated Data

The deterministic bivariate polynomial used in the interpolation introduces bias into the correlation estimation. Such bias is concentrated at the scale of the gage spacing, approximately 1 km, because the interpolation is applied to clusters of three proximal gages. The spatial extent of this bias can be seen clearly by examining the interpolated field resulting from a 30 mm pulse of rainfall at gage 29 as shown in the upper portion of Figure 7.3.3. In this case at least, the "smearing" of the impulse falls below 10% of the impulse strength at about 2.5 km. The effect of interpolation on the correlation structure is shown in the lower portion of Figure 7.3.3 where the correlation function of the interpolated impulsive storm is presented. We see that the bias is contained within $\nu = 2.5$ km.

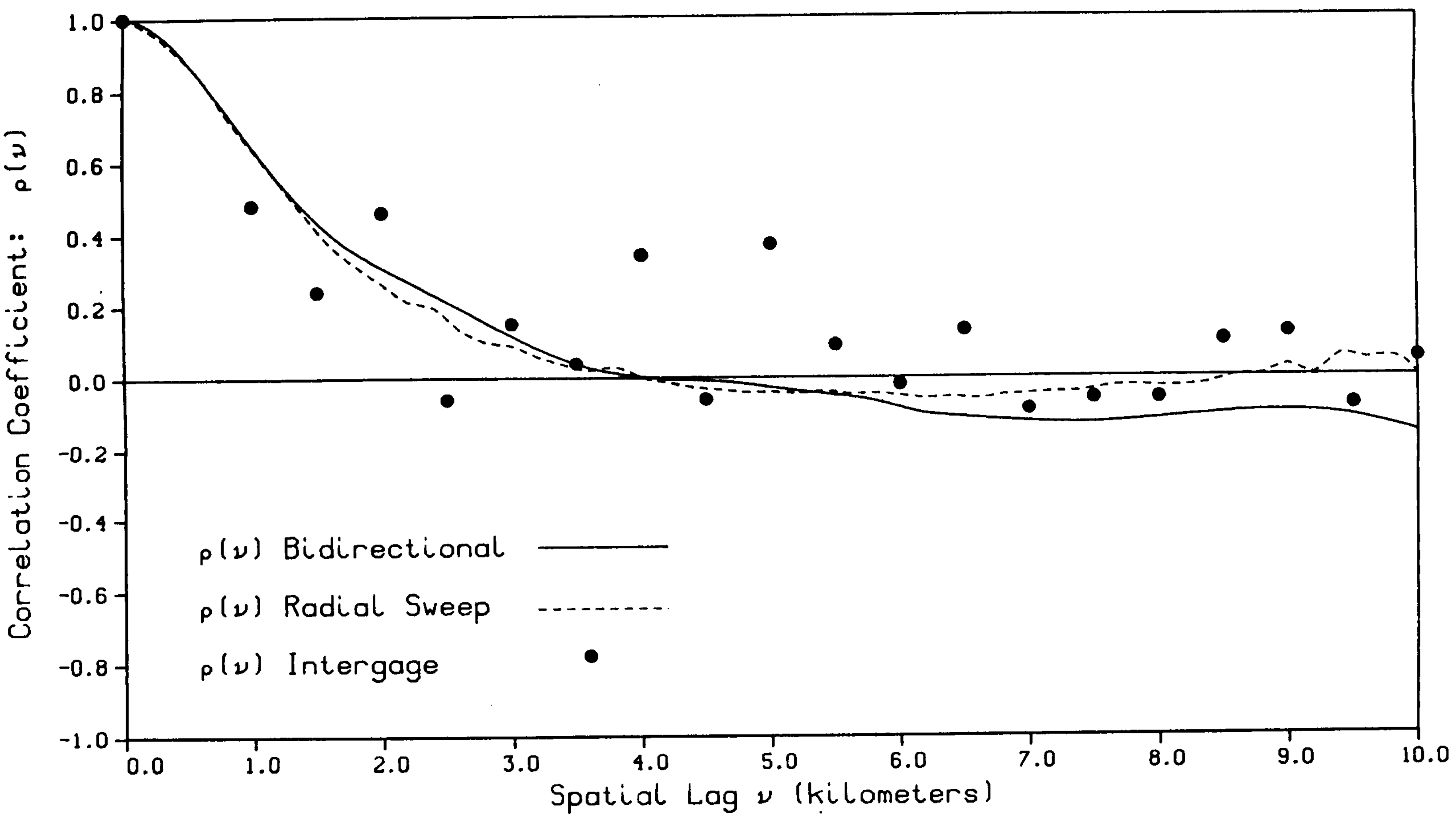

Fig. 7.3.1 Estimation of the 22 June 1970 Storm Day Spatial Correlation

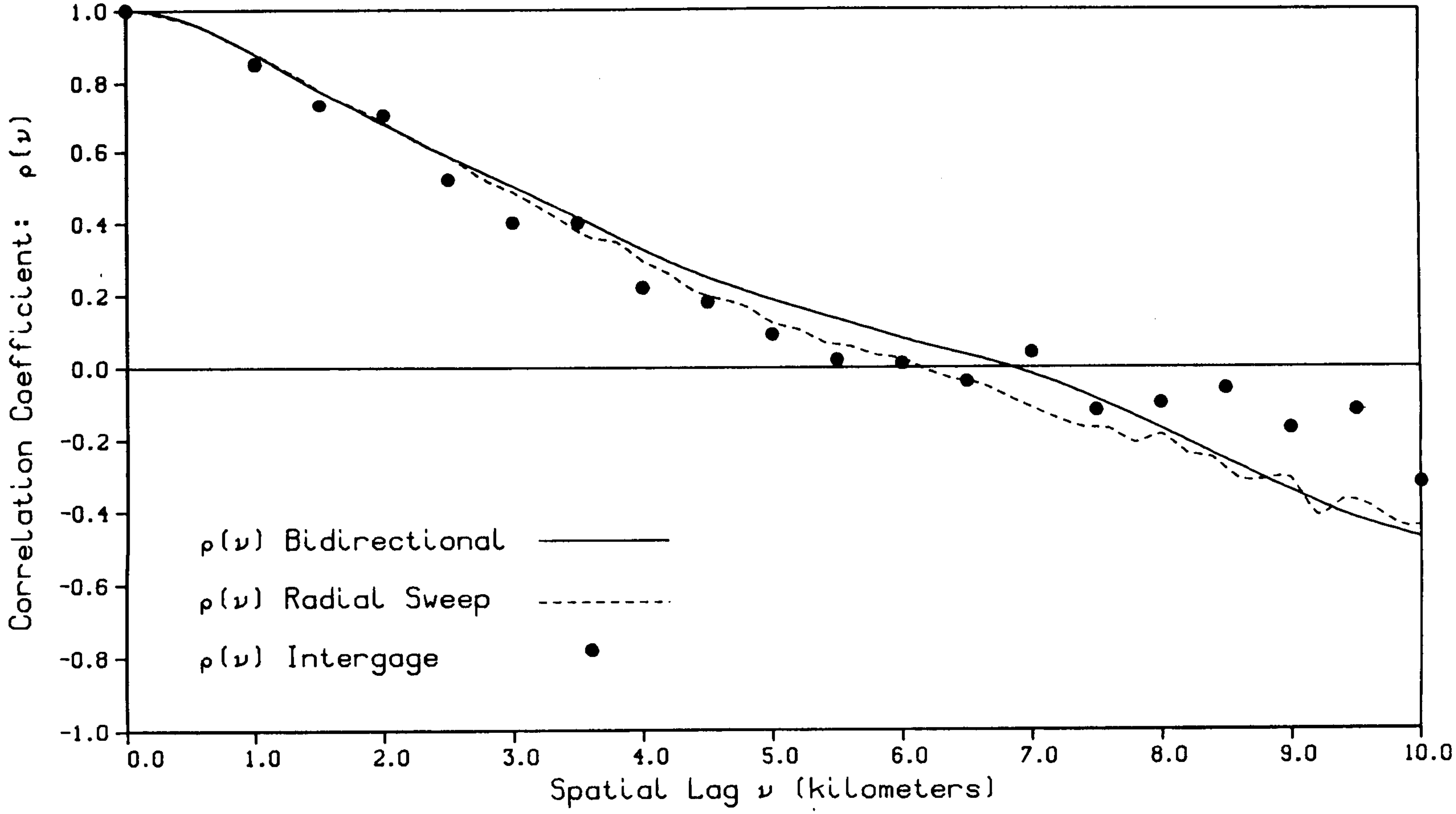

Fig. 7.3.2 Estimation of the 24 July 1970 Storm Day Spatial Correlation

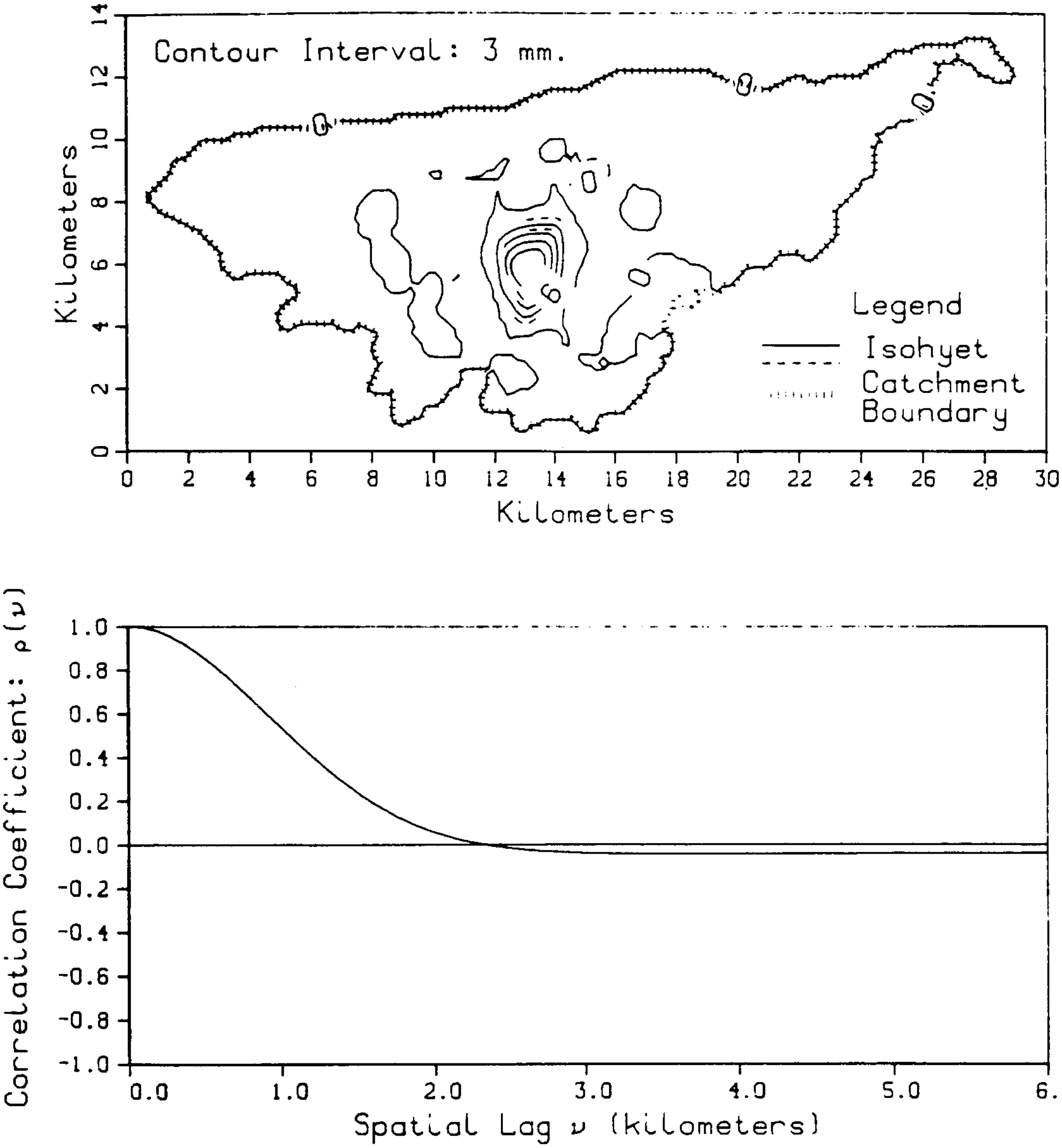

Fig. 7.3.3 Gage 29 Unit Impulse

The primary correlation information resides at what is called [Mejia and Rodriguez-Iturbe, 1973] the "characteristic correlation distance" which is the expected distance between two independent randomly chosen points within the particular catchment. Using the method of Mejia and Rodriguez-Iturbe [1973] and approximating the Walnut Gulch catchment by a 2:1 rectangle of 154 km^2 its characteristic correlation distance is 7.0 km.

In sampling the finite random field for correlation estimation, the maximum covariance lag is limited by the dimension of the field, as bias is introduced by declining sample size with increasing lag. Studies of this problem led to a limit of 6 km for use in parameter estimation. For purposes of comparing methods of estimating $\rho(\nu)$ we will extend this calculation to 10 km.

Sampling from the interpolated random field (henceforth simply referred to as the random field) requires several considerations. The first constraint in designing an algorithm to perform this analysis is to assure that sampling is confined to the inside of the basin boundary. The second constraint is to ensure that each pair of node points in the random field is sampled only once. This aspect will be be examined in detail in the following sections. The third consideration involves the amount of computational time required for a particular scheme. This final aspect has no definite rule and must be resolved subjectively.

Henceforth, we define the "pivot point" as the principal node under consideration. The "secondary point" lies a lag distance ν km away from the pivot point. Only the nodes inside the basin boundary belong to the

Walnut Gulch random field. These nodes are the only potential pivot and secondary point candidates. All other nodes have been assigned a value of -1.0 mm. and are excluded from consideration.

7.3.3 Sampling the Walnut Gulch Random Field for Correlation

Four techniques to determine the spatial correlation of the random field are presented. A node by node technique is presented in 7.3.4, a radial sector technique is presented in 7.3.5, a radial sweep technique is presented in 7.3.6 and a bidirectional technique is presented in 7.3.7. Each algorithm is especially designed with the Walnut Gulch random field in mind.

Each algorithm begins the examination of the interpolated random field starting with the node located in the first column and the last row of the matrix (southwest corner). Because this node lies outside the random field, (i.e. has a point depth of -1.0 mm., the algorithm shifts to the next node to the right. The algorithm is designed to scan first from west to east along a row before beginning to search for the first node of the random field in the next row (from south to north). Nodes that fail to pass the logical filter are skipped until the algorithm comes to rest at the first node which is a member of the random field.

The first node of the random field is located in row 5 column 63 of the coarse mesh. This node becomes the first "pivot point". Each of the remaining 4004 nodes then become "secondary points" depending on the particular algorithm. Following the calculations involving all pairs of the given pivot point and set of secondary points, the algorithm moves to the next node, located to the right of the pivot point. This node becomes the new pivot point. Because the old pivot point and the new pivot point (an old secondary point) pair has already been examined, it is necessary to exclude the old pivot point from becoming a member of the set of new secondary points. The algorithm is designed to scan the random field of secondary points beginning with the secondary point to the right of the pivot point, continuing along the same row to the last column (eastern edge) of the random field. The algorithm then begins an examination of the remaining secondary points in the row above (to the north) the pivot point, and scans west to east. Depending on the particular technique, the details differ a bit, but a row by row examination is completed until the set of secondary points (with respect to each pivot point) in the random field is exhausted.

Once again we point out that for large lags, a bias is introduced that favors the center of the finite random field. By limiting the maximum lag distance the magnitude of this bias can be contained but is present at all lags $\nu > 0$. This bias arises primarily within a region near the catchment boundary which is proportional to the magnitude of ν. The remaining area of the catchment is the preferred domain for correlation estimation. The estimates in this report are only partially restricted to this area, and we attempt to limit the resulting bias by a subjective choice of maximum spatial lag. This issue will be discussed later in the chapter.

In summary, the algorithm only examines the field of secondary points in the same row and columns to the right of each pivot point and then in all columns and rows above the pivot point. Therefore, the algorithm never examines nodes to the left of the pivot point or in rows below because these nodes have already been pivot points or else lie outside the random field. In this way, we are assured that each pair of pivot and secondary points are examined only once.

7.3.4 Node by Node technique

One approach to sampling the random field for the spatial correlation is the node by node technique. The basic approach is to determine the estimation at a particular lag by considering the pivot point and each of the remaining secondary points in the random field. Certainly this approach results in the best estimation of the covariance function; however, its obvious disadvantage is the tremendous degree of computational effort.

7.3.5 The Radial Sector Technique

Another approach to sampling for the spatial correlation is the radial sector technique. In this technique, the set of secondary points is subdivided into radial sectors which emanate from the pivot point. A subset of secondary points lies inside an element of the sector having area $r\Delta r\Delta\theta$. The values of the secondary point depths within this area are averaged and this single value becomes the secondary point value in the covariance summation at a lag distance of $r + \Delta r/2$ from the pivot point. Unless the incremental angle $\Delta\theta$ and the width of the slice Δr are small, the data are significantly smoothed at larger lag distances from the pivot point. If these conditions are met however, the required number of calculations approaches that of the node to node technique described above.

7.3.6 The Radial Sweep Technique

We developed and investigated a technique which we believe is unique. At the first pivot point (row 5 column 63), the algorithm extends a ray from the pivot point to the node in the last column of the same row (eastern edge) of the matrix. The algorithm then determines the exact number of secondary points along this ray. Next, the distance separating a pivot point-secondary point pair is calculated and rounded up or down to the nearest 200 meters. Detrending by the mean is then followed by the other calculations necessary to contribute to the appropriate lag distance covariance and variance estimate summations. Following this step, another ray is extended from the pivot point to the node in the last column of the matrix, but one row above the pivot point. The same procedure is followed as before.

For each pivot point in the random field a series of rays is created which first sweeps the last column (east side) of the matrix beginning at the same row as the pivot point upward to the last column of the top row (northeast edge). A series of rays then sweep the first row (north side) of the matrix from the second to last column, to the first (east to west) and then from the second row, first column of the mesh finally to the row above the pivot point (north to south). The algorithm then shifts to the node to the right of the pivot point. This point then becomes the new pivot point and the process is repeated.

Because the radial sweep is always less than 180°, duplication of pairs is not possible. Furthermore, no smoothing is performed on any of the values in the random field. The lag distance between the pivot point and each secondary point is rounded up or rounded down to the nearest 200 meters. The correlation function determined in this manner is presented for the two representative storm days in Figures 7.3.1 and 7.3.2 where it can be compared with that from the gage observations. We note as expected that the (interpolated) radial sweep estimate is slightly higher than the gage estimate (at least for $\nu \leq 6$ km); however, the agreement is satisfactory.

The computer program that was developed to perform this technique is called ALLCORR.FORTRAN and is described in Chapter 12. Despite its advantage in precision, this approach is considered computationally excessive and was abandoned.

7.3.7 The Bidirectional Technique

The technique developed and used in this study is called the bidirectional technique. This approximate technique is computationally fast, and compares reasonably well with the results from the more exact radial sweep technique described above. In this approach, a ray is extended from the pivot point along the same row to the last column of the matrix (west to east). One by one, each pivot-secondary point pair is examined and the various calculations necessary for contribution to the appropriate covariance and variance estimates are performed. The lag distance separating the pair is always known and unnecessary to calculate, or round up or round down. Each pivot point in the random field follows the same procedure of extending a ray along the same row to the eastern edge of the matrix.

A row by row examination is completed for all pivot points and then a similar column by column examination is begun. A ray is extended from the pivot point to the top row of the matrix (south to north). At each lag, the pivot and secondary point pair contribute to the same appropriate covariance summations begun during the row by row examination.

The basic algorithmn to determine the covariance at each lag is the following. Let Y = y(i,j) represent the rainfall depth at the node point located in column i and row j of the random field. Let I and J equal the total number of columns and rows respectively in the rectangular matrix. Provided that neither y(i,j) or y(i+k,j+k) equal -1.0 mm., the covariance at lag k is then estimated by

$$\mathrm{COV}[Y(i,j),Y(i+k,j+k)] = \frac{1}{n}\sum_{i=1}^{I-k}\sum_{j=1}^{J-k}\left(y(i,j)-E[Y]\right)\left(y(i+k,j+k)-E[Y]\right) \tag{7.3.9}$$

for $k \geq 0, 0.2, 0.4, \ldots,$ max.km.

where, as before, E[Y] equals the expected value of total rainfall depth in the entire random field and $n = n(\nu)$ which equals the number of product pairs at lag ν. It is important to understand how the spatial correlation was estimated.

$$\rho(\nu_k) = \frac{\sum_{i=1}^{I-k}\sum_{j=1}^{J-k}\left(y(i,j)-E[Y]\right)\left(y(i+k,j+k)-E[Y]\right)}{\left\{\sum_{i=1}^{I-k}\sum_{j=1}^{J-k}[y(i,j)-E[Y]]^2\right\}^{1/2}\left\{\sum_{i=1}^{I-k}\sum_{j=1}^{J-k}[y(i+k,j+k)-E[Y]]^2\right\}^{1/2}} \tag{7.3.10}$$

The correlation function estimated by this approximation is presented for the two representative storm days in Figures 7.3.1 and 7.3.2. We note that the bidirectional method closely approximates the more exact radial sweep method and is an acceptable approximation of the gage correlation at least to lags of 6 km.

7.4 Sampling for the Variance Function

7.4.1 Introduction

The variance function is a measure of the reduction of the point variance under local averaging [Vanmarcke, 1983]. In one dimension the average over T of Y(t) is $Y_T(t)$ or simply Y_T.

$$\mathrm{VAR}(Y_T) \equiv \sigma_T^2 = \gamma(T)\sigma^2 \tag{7.4.1}$$

where

σ^2 = variance of Y(t), and

$\gamma(T)$ = variance function of Y(t)

$\gamma(T)$ is dimensionless and has the following properties:

$$\gamma(T) \geq 0 \tag{7.4.2}$$

$$\gamma(0) = 1 \tag{7.4.3}$$

$$\gamma(-T) = \gamma(|T|) = \gamma(T) \tag{7.4.4}$$

The variance function $\gamma(T)$ is related to the correlation function $\rho(\tau)$ by [Vanmarcke, 1983]

$$\gamma\ (T) = \frac{1}{T^2} \int_0^T \int_0^T \rho(t_1 - t_2)\ dt_1 dt_2$$

$$= \frac{2}{T} \int_0^T \left[1 - \frac{\tau}{T}\right] \rho(t)\ d\tau \tag{7.4.5}$$

Similar characteristics define the variance function $\gamma(X_1, X_2)$ in the two-dimensional plane.

For isotropic random fields, averaging over the rectangular area $A = X_1 X_2$ gives [Vanmarcke, 1983, p. 242]

$$\gamma(A) = \gamma(X_1,X_2) = \frac{1}{X_1X_2}\int_0^{X_1}\int_0^{X_2}(1 - \frac{x_1}{X_1})(1 - \frac{x_2}{X_2})\rho(x_1,x_2)\ dx_1dx_2 \qquad (7.4.6)$$

7.4.2 Sampling the Random Field for the Variance Function

From the definitions of the variance function $\gamma(X_1,X_2) = \gamma(A)$ presented in section 7.4.1, we wish to sample the Walnut Gulch random field y(i,j) at discrete lag distances. By definition,

$$\gamma(A) = \frac{\sigma_A^2}{\sigma^2} \qquad (7.4.7)$$

where $\sigma^2 = \mathrm{VAR}[Y(\underline{x})]$, the point variance of the total depth random field and σ_A^2 equals the variance of Y_A, the storm depth locally averaged over a finite element of area A.

In order to simplify the algorithm design, only symmetrical lag distances in the i and j directions are used, creating square elements of dimension $X_1,X_2 = L$. The algorithm also overlaps these elements in the following way. Consider an element (the "first element" here) of area A in the random field. The western edge of the element next to, and to the right (east) of this element is always 200 meters from the western edge of the first element. Similarly, the southern edge of an element located to the north of the first element, always lies 200 meters from the southern edge of the first element. There is no diagonal overlapping of elements.

The numerical integration of eq. 7.4.6 for positive lengths only is accomplished by simply averaging the point rainfall depth of the random field nodes located within the area of element k. Thus

$$Y_A(X_1,X_2) = \frac{1}{X_1X_2}\int_0^{X_1}\int_0^{X_2} Y(x_1,x_2)\, dx_1 dx_2 \qquad (7.4.8)$$

provided that y(i,j) does not equal -1.0 mm., eq. 7.4.8 may be expressed for square elements as

$$Y_A(X_1,X_2) = \frac{1}{(L+1)^2}\sum_{i=n}^{n+L}\sum_{j=n}^{n+L} y(i,j) \qquad (7.4.9)$$

where L=0.0,0.2,0.4,..., 6 km.

In order to simplify notation we will let

$$Y_A = Y_A(X_1,X_2) \qquad (7.4.10)$$

therefore, at a length of L kilometers, the corresponding variance function $\gamma(A) = \gamma(L^2)$ is defined by

$$\gamma(A) = \frac{VAR(Y_A)}{VAR(Y)} \qquad (7.4.11)$$

where as before, VAR(Y) is the variance of the point depth and VAR(Y_A) equals the variance of the average over A of the point depth.

The basic algorithm is similar in structure to the spatial correlation function algorithm described in section 7.3. This algorithm searches the matrix for the first node of the random field, beginning column by column in the last row before starting to search in the next to last row, and so forth. At L = 0.0 (A = 0.0 km^2), VAR(Y_A) equals VAR(Y), and the random field is sampled as described in section 7.2.

At length L = 0.2 (A = 0.04 km^2), the search stops at row 5 column 63 of the coarse mesh. This node becomes the primary point (equivalent to the pivot point in the spatial correlation algorithm). Three other nodes comprise the set of potential secondary points in this first element of the random field. The first secondary point is the node located in the next column (200 meters to the right of the primary point), the second is located in the next row, same column (200 meters to the north) as the primary point, and the third is in the row above and in the same column as the first secondary point (northwest of the primary point). During the summation of eq. 7.4.9, each of the secondary points is passed through the logical filter. If any secondary point fails the test, the algorithm discontinues the summation. The algorithm then moves to the next node in the same row as the first primary point until it locates the second primary point and begins the logical examination and integration process again. The average depth of the four nodes at L = 0.2 km. becomes the depth of this element of A = 0.04 km^2.

More elements are created along the same row, and then the algorithm shifts to the row immediately above this one (200 meters to the north) and continues the process as before column by column (west to east). The algorithm moves row by row in this fashion until the random field is exhaused of elements of this size area. Next, the variance of the element depth is calculated and subsequently, the value of the variance function for elements of this area.

The algorithm then increases the element size to L = 0.4 km. (A = 0.16 km^2). For each element in the random field, there must be one primary point and 8 secondary points. Again, the algorithm searches for the first primary point which is located in row 5 column 63 of the coarse mesh.

The elements are always square in shape, and the size is increased in increments of 0.2 km of length during each scan of the random field for the variance function at this size of element. Ultimately, the maximum length was carried out to 6 km. (A = 36.0 km^2). An element of this size requires the primary point and 960 secondary points. At this length, (L = 6.0 km), there are about 100 elements in the random field, and the algorithm is complete.

In order to further clarify the algorithm, consider a random field of dimension I × J. The average depth of a single element Y_A is described by eq. 7.4.9. The expected value of element depth $E(Y_A)$ is then

$$E(Y_A) = \frac{1}{n_A} \sum_{k=1}^{n_A} Y_{A_k} \qquad (7.4.12)$$

where n_A equals the number of elements of area A in the random field. The variance of the depth of the elements of area A is thus

$$VAR(Y_A) = \frac{1}{n_A - 1} \sum_{k=1}^{n_A} Y^2_{A_k} - E^2(Y_A) \qquad (7.4.13)$$

The variance function $\gamma(A)$ at this lag is described by eq. 7.4.11.

7.4.3 Observations about Sampling for the Variance Function

During the course of the storm day analysis described in Chapter 10, we observed that the variance function often rose to values greater than one at small element areas before declining and approaching the zero asymptote for large areas. This behavior was unexpected in light of the theoretical definition of the variance function.

On closer examination, we deduce from the following example, that such an occurrence is possible. Consider the small random field in Figure 7.4.1 composed of four elements at lag 1 and nine nodes.

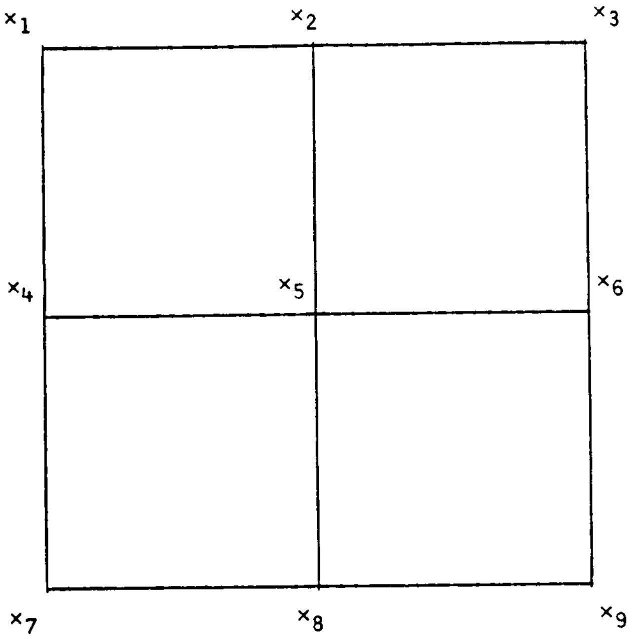

Fig. 7.4.1 A Simple Illustrative Grid Mesh

Let $\overline{X}$ = E(Y) be the average node depth

$$\overline{X} = \frac{x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9}{9} \qquad (7.4.14)$$

let $\overline{\overline{X}}$ = E(Y_A) be the average of the element averages

$$\overline{\overline{X}} = \frac{\frac{x_1 + x_2 + x_4 + x_5}{4} + \frac{x_2 + x_3 + x_5 + x_6}{4} + \frac{x_4 + x_5 + x_7 + x_8}{4} + \frac{x_5 + x_6 + x_8 + x_9}{4}}{4}$$

(7.4.15)

From these expressions we are able to solve for VAR(Y) and VAR(Y_A). In order for the variance function to assume a value greater than one, VAR(Y_A) must be greater in value that VAR(Y). This will occur whenever a > b where

$$\begin{aligned} a = {} & 17{,}392X_5^2 + 2{,}812(x_2^2+ x_6^2+ x_8^2) \\ & + 350(x_1x_3+ x_1x_7+ x_1x_9+ x_3x_7+ x_3x_9+ x_7x_9) \\ & + 188(x_1x_2+ x_2x_3+ x_2x_7+ x_2x_9+ x_1x_6+ x_3x_6) \\ & + 188(x_6x_7+ x_6x_9+ x_1x_8+ x_3x_8+ x_7x_8+ x_8x_9) \end{aligned} \tag{7.4.16}$$

and

$$\begin{aligned} b = {} & 833(x_1^2+ x_3^2+ x_7^2+ x_9^2) + 748)x_2x_5+ x_5x_6+ x_5x_8) \\ & + 136(x_1x_5+ x_3x_5+x_5x_7+ x_5x_9) \\ & + 136(x_2x_6+ x_2x_8+ x_6x_8) \end{aligned} \tag{7.4.17}$$

We should remember that for computational economy the variance function is developed here for square areas. In applying the definition to oddly-shaped catchments such as Walnut Gulch there will be "uncovered area" near the catchment boundaries that results in a biased estimate of $\gamma(A)$ (see Vanmarcke [1983, p. 254]).

CHAPTER 8

Storm Day Observations: Sampling from the Fine Grid Mesh Random Field

8.1 Introduction

The spatial distribution of rainfall is of great interest in surface hydrology and is the central objective of this study. It will dictate the degree of local surface flooding, infiltration and available moisture for evaporation.

Expressed as a fraction, the spatial distribution of rainfall depth is described by the wetted fraction of catchment area

$$\frac{A_{cw}(Y \geq y)}{A_c} \qquad (8.1.1)$$

where A_c is the catchment area, and A_{cw} is the wetted catchment area.

Of first order interest is the fraction of total area not wetted by the precipitation event, i.e. the dry fraction. The dry fraction of total area is expressed by

$$\frac{A_{cd}}{A_c} \qquad (8.1.2)$$

where A_{cd} is the dry catchment area.

Following testing with the coarse mesh grid (200 meter resolution), a fine grid mesh was adopted for this aspect of the study. As discussed in Chapter 6, the resolution of the fine mesh is 100 meters. The fine mesh random field (nodes inside the limit of the watershed) is composed of 15,905 nodes. The following sections describe the techniques used to evaluate equations 8.1.1 and 8.1.2.

8.2 Sampling for the Spatial Distribution of Rainstorm Depth

A computer program called STORMWET.FORTRAN is developed to determine the spatial distribution of total storm day depth. The program is described in Chapter 12.

Following the creation of the interpolated total depth surface and the implementation of the fine mesh boundary filter, each node is passed through the minimum depth filter of y = 0.01 mm described in Chapter 6.

The basic algorithm begins to search for the first node of the random field by starting in the first column, last row of the matrix as is the case in sampling for the moments, spatial correlation, and variance function. The first node of the random field is located in row 10 column 127 of the fine mesh matrix. As was the case in the description of the variance function algorithm, this node is called the primary point. In the analysis of the spatial distribution of total rainfall, the element length is a constant 100 meters.

The algorithm then inspects the 3 secondary points. The first secondary point is located in the same row, next column (eastward) to the primary point. The second secondary point is located in the prior row, same column (to the north) of the primary point. The third secondary point is located in the prior row, same column as the first secondary point (northeast of the primary point). Provided that each secondary point passes the logical filter which examines each in turn, the rainfall depth of the primary point and the three secondary points are averaged.

This value represents the depth of the first element of area (0.01 km^2). The algorithm then looks to the right of the first primary point (same row) until it locates the next member of the random field.

On completion of the creation of the elements in the first row of the random field, the algorithm then moves to the prior row (north) and begins the column-by-column creation of the elements along that row. The algorithm is complete when all rows of the random field have been examined (south to north). In the fine mesh random field, a total of 15,421 elements are created.

In order to clarify the scheme, consider a rectangular fine mesh matrix of (M × N) dimension. Y is the 1 × k matrix of element depth and the depth of the finite element k is given by

$$Y_k = \frac{1}{4} \sum_{i=m}^{m+1} \sum_{j=m}^{m+1} y(i,j), \qquad y(i,j) \neq -1 \text{ mm} \qquad (8.2.1)$$

$$k = 1,2, \ldots, 15421$$

Next the elements are sorted into groups according to the depth of the individual elements after finding the maximum depth among all of the elements. This sorting procedure performs the areal integration. The elements are sorted into groups of elements by integer millimeter depths and then the algorithmn divides the number of elements in each group by the total number of elements in the basin. Two groups receive special treatment. The first includes all the elements with a depth equal to zero mm. These elements represent all the "dry" elements. The remaining elements are all a part of the set of "wet" elements which have a depth greater than zero mm.

Because the areal integration is the only means of calculating the area within the Walnut Gulch watershed, this procedure is an effective way to determine the accuracy of the logical boundary filter. According to the Agriculture Research Service, the area of the basin is 150 km^2. The numerical integration scheme results in an area of 154.21 km^2. We conclude that the error of 4 km^2 is due either to inaccuracies of the base map watershed boundary provided by the ARS, or to approximations introduced by the numerical boundary.

8.3 Sensitivity Analysis of the Sampling for the Spatial Distribution of Total Rainfall Depth

In chapter 6, we described the minimum depth filter through which all the nodes in the random field were passed prior to any analysis. This filter assigns a depth of 0.0 mm to any node with a value less than or equal to 0.01 mm total rainfall depth. Certainly, 0.01 mm is at the extreme lower limit of rainfall measurement and is much less than the 0.01 inch resolution of the tipping bucket gage.

The level of this filter has an impact on the estimate of the fraction of basin area that remains dry during a storm day. Furthermore, the resolution of the mesh also plays a part as mentioned in the preceding section. The mesh resolution also has a small effect on the estimated spatial distribution of storm area wetted to depths greater than 1.0 mm, but not enough to be of any consequence.

In order to investigate these sensitivities, we examined the dry fraction of total area A_{cd} for the 22 June 1970 and 24 July 1970 representative storm days. Tests were conducted using combinations of four differently sized finite elements and four minimum depth filters.

The first element size tested is a square element from the coarse mesh having an area of 40,000 m^2 or 0.04 km^2. This element depth is the average depth of the four nodes located at the corner of each of the elements. The second element size tested is a triangular shaped element also from the coarse mesh. Each pair of triangular elements is a single coarse mesh square element divided in half. Each of these elements has an area of 0.02 km^2. The depth of each element is the average of the node depths at the vertices of the triangle.

The third element size tested is the fine mesh square element described in Section 8.2. The area of each of these elements is 0.01 km^2, and the depth of the element is the average depth of the four nodes at its corners. The fourth element size tested is a triangular element. Two of these elements come from each of the square fine mesh elements. As is the case for the coarse mesh triangular elements, the element depth is the average depth of the nodes at the vertices of each triangle. The area of these elements equals 0.005 km^2.

Four minimum depth filters are examined. The first is the 0.01 mm filter previously described, the others are 0.1 mm, 0.05 mm, and zero.

The results of this analysis are presented in Table 8.3.1 and Table 8.3.2 for the 22 June 1970 and the 24 July 1970 storm days.

Table 8.3.1

Sensitivity of A_{cd} to Minimum Depth Filter
and Finite Element Dimension
Storm Day 22 June 1970

	Element Area (km^2)			
Filter (mm)	0.04	0.02	0.01	0.005
< 0.1	0.783	0.788	0.798	0.800
≤ 0.05	0.754	0.759	0.772	0.774
≤ 0.01	0.692	0.698	0.714	0.718
0.0	0.497	0.508	0.535	0.541

Table 8.3.2

Sensitivity of A_{cd} to Minimum Depth Filter
and Finite Element Area Dimension
Storm Day 24 July 1970

	Element Area (km^2)			
Filter (mm)	0.04	0.02	0.01	0.005
<0.1	0.147	0.153	0.159	0.162
≤ 0.05	0.138	0.143	0.149	0.151
≤ 0.01	0.109	0.115	0.122	0.126
0.0	0.092	0.098	0.105	0.109

Table 8.3.3 shows the sensitivity of the estimated storm day dry catchment area with respect to the dimension of the finite element area size for each of the minimum depth filters. $A_{cd}(A = 0.005 \text{ km}^2)$ equals the dry catchment area as estimated by using the 0.005 km^2 finite element. $A_{cd}(A = 0.04 \text{ km}^2)$ represents the estimated dry catchment area from using

0.04 km^2 finite element. The dimensionless ratio of the former divided by the latter is presented in this table for each minimum depth filter.

Table 8.3.3

A_{cd} Sensitivity to Finite Element Dimension

Filter (mm)	A_{cd} (A = 0.005 km^2)/A_{cd} (A = 0.04 km^2)	
	22 June 1970	24 July 1970
<0.1	1.02	1.10
$\leq$0.05	1.03	1.04
$\leq$0.01	1.04	1.16
0.0	1.09	1.18

Table 8.3.4 illustrates the sensitivity of A_{cd} with respect to the magnitude of the minimum depth filter. A_{cd} ($\leq$ 0.01) equals the estimated dry catchment area in each storm day when using this filter. A_{cd} (0.0 mm) equals the estimated dry catchment area from using absolute zero as the cutoff. The dimensionless ratio of the former divided by the latter is presented for the two storm days.

Table 8.3.4

A_{cd} Sensitivity to Minimum Depth Filter

Element Area (km^2)	A_{cd}($\leq$ 0.01 mm)/A_{cd} (0.0 mm)	
	22 June 1970	24 July 1970
0.04	1.39	1.18
0.02	1.37	1.17
0.01	1.33	1.16
0.005	1.33	1.16

These results suggest that the sensitivity of A_{cd} is storm day dependent. For the "sparse" 22 June 1970 storm day, A_{cd} is most strongly influenced by the magnitude of the minimum depth filter. On the other hand, we see that there is little variation with the magnitude of the filter or with the size of the element for the 24 July 1970 storm day.

CHAPTER 9

Storm Day Data Processing Computer Programs

9.1 Storm Day Data Processing

There are several steps required to process a single storm day's station observation data. This chapter outlines the procedures and the computer programs necessary to execute the process from start to finish. Details concerning file manipulation, linking with the numerical and graphical package libraries will be discussed in Chapter 12.

Following the implementation of the computer program entitled COORD.FORTRAN, the processing of a single storm day is ready to begin.

9.1.1 STRMSORT.FORTRAN

The first program to be executed during a single storm day processing is called STRMSORT.FORTRAN. The purpose of this computer code is to gather together the 93 station observations for a single storm day from one of several files (depending on the year) and place them into a single file for reading.

9.1.2 STORMDAY.FORTRAN

The second computer program to be implemented is called STORMDAY.FORTRAN. This program accomplishes all of the coarse mesh computational tasks outlined in Chapter 7. First, STORMDAY.FORTRAN reads the file containing the 93 station observations for this storm day, created by STRMSORT.FORTRAN. The code then calculates the first two moments of the the station observations. Next, the coarse mesh total rainfall depth surface is created by the bivariate surface interpolator, after which, the numerical boundary filter is imposed. Following the passing of the random field through the minimum depth filter, The first three moments of the random field are determined. This step is followed by the calculations required to determine the spatial correlation and the variance function. These steps are the last computations performed by this program.

The next portion of STORMDAY.FORTRAN involves graphics preparation and writing the computational results to a file. The first step is the implementation of two special filters used solely for the graphical presentation of the results. The first filter creates the watershed boundary outline, and the second filter assures that the zero depth isohyet will be drawn. Next, the program writes the first two moments, the coefficient of skewness, the spatial correlation and the variance function computational results to a file.

STORMDAY.FORTRAN then begins the graphics routines. These result in a single plot file which contains: the storm day total depth isohyet map; the spatial correlation plot and the variance function plot.

9.1.3 STORMWET.FORTRAN

The third program to be executed during a single storm day processing is called STORMWET.FORTRAN. The sole purpose of this code is to determine the spatial distribution of total rainfall depth in the random field. STORMWET.FORTRAN is the only program which creates and samples from the fine mesh surface as described in Chapter 8.

STORMWET.FORTRAN first reads in the 93 station observations from the file created by STRMSORT.FORTRAN. Next, the fine mesh total depth surface is created by the bivariate interpolator followed by the numerical basin boundary filter. The random field is then passed through the minimum depth filter, followed by the creation of the 0.01 km^2 finite total depth elements of the random field.

The code then screens all the total depth elements in order to find the element with the greatest depth. Next, the elements are sorted according to the element's value, and that depth is changed from a real number to an integer value. There is now a group of elements with values exactly equal to zero, a group of elements with depth greater than zero, and groups of elements with various depths ranging from 0 to 1 mm at one extreme to the group ranging from 0 to the maximum among all the elements in the random field at the other extreme. The total number of elements in each group is divided by the total number of elements possible in the random field and subtracted from 1.0. This value constitutes the fraction which results in the spatial distribution of total rainfall depth within the basin. These results are then written to a file.

9.1.4 TABLE.FORTRAN

The fourth and final program to be executed during a single storm day processing is called TABLE.FORTRAN. The purpose of this code is to merge the file created by STORMDAY.FORTRAN with the file created by STORMWET.FORTRAN. TABLE.FORTRAN first reads in these two files and then writes these combined results to two different files. One file results in a data table suitable for presentation and the second contains all the same data but formatted into a condensed version. The first file may be then printed by the analyst should he so chose. The second file is transferred by the analyst (not computer) to another file which is an archive file. This archive file is a single year of storm day processing results and will be discussed in Chapter 10.

9.1.5 Further comments about storm day data processing

Depending on the particular storm day, the total processing time requires between 20 and 30 central processor unit minutes to complete. The most time consumming aspect of the procedure is the creation of the isohyet plot by the graphics package, followed by the calculations necessary to determine the variance function. Both of these routines are part of the STORMDAY.FORTRAN computer program. The execution time by STORMWET.FORTRAN is completely dependent on the magnitude of the maximum depth element in the random field. During the course of this study, storm day data processing was especially hindered by as yet unresolved "bugs" in the proprietary graphics package. These include the necessity to "restart" the execution of the STORMDAY.FORTRAN program when the graphics routines are drawing the watershed boundary in the isohyet plot. Also, if the depth of the random field has fairly significant gradients (or perhaps for some other reason), the graphics package will enter what appears to be an infinite loop if the contour interval is "too small". Because of these two problems, it is unfortunately not possible to execute a series of storm day processings without user intervention.

Initially, BALANCE.FORTRAN yields 431 storm days for the eight summer seasons. The graphics package invoked during the execution of the STORMDAY.FORTRAN program refused to complete the isohyet plots of three of these. The 93 station observation input file for each of these days (created by STRMSORT.FORTRAN) was examined and we found that for each, only one or two raingages had a total depth of 1 mm. These three storm days were subsequently rejected and so finally a total of 428 storm days were successfully processed.

The processing results of the 22 June 1970 and 24 July 1970 storm days are presented in Figs. 9.1.1 and 9.1.2. and in Tables 9.1.1 and 9.1.2. The spatial distribution of total storm depth for each of these two storm days is illustrated in Figs. 9.1.3 and 9.1.4 respectively. These plots are from information presented in Tables 9.1.1 and 9.1.2.

The complete results of the 428 storm day spatial analyses are presented in the two volume set by Fennessey et al. [1986]. Presented are: total depth isohyets at 2 mm contour intervals, the moments of point depth, A_{cd}/A_c, A_{cw}/A_c, the spatial correlation, the variance function and the spatial distribution of total storm depth.

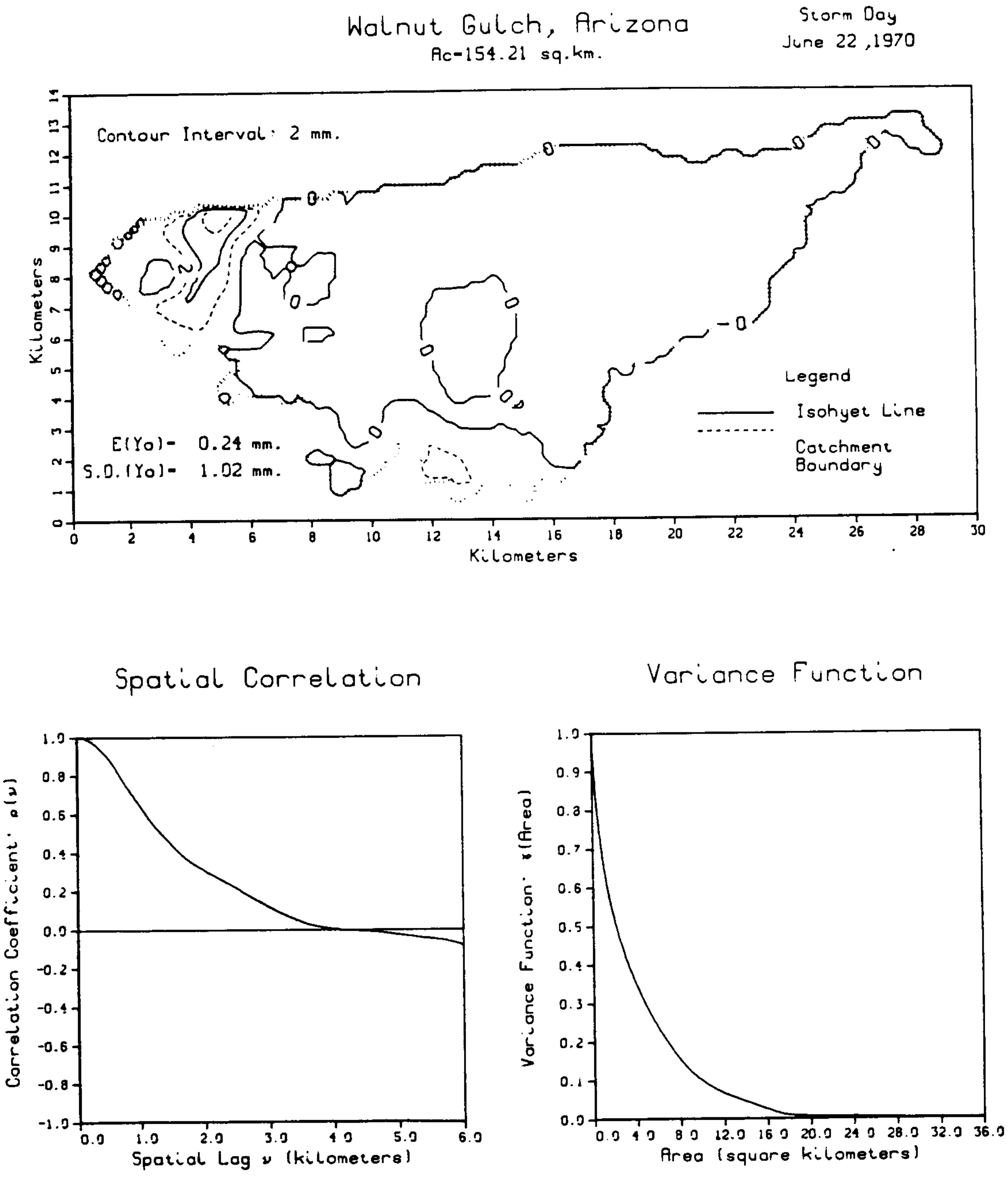

Fig. 9.1.1 22 June 1970 Storm Day

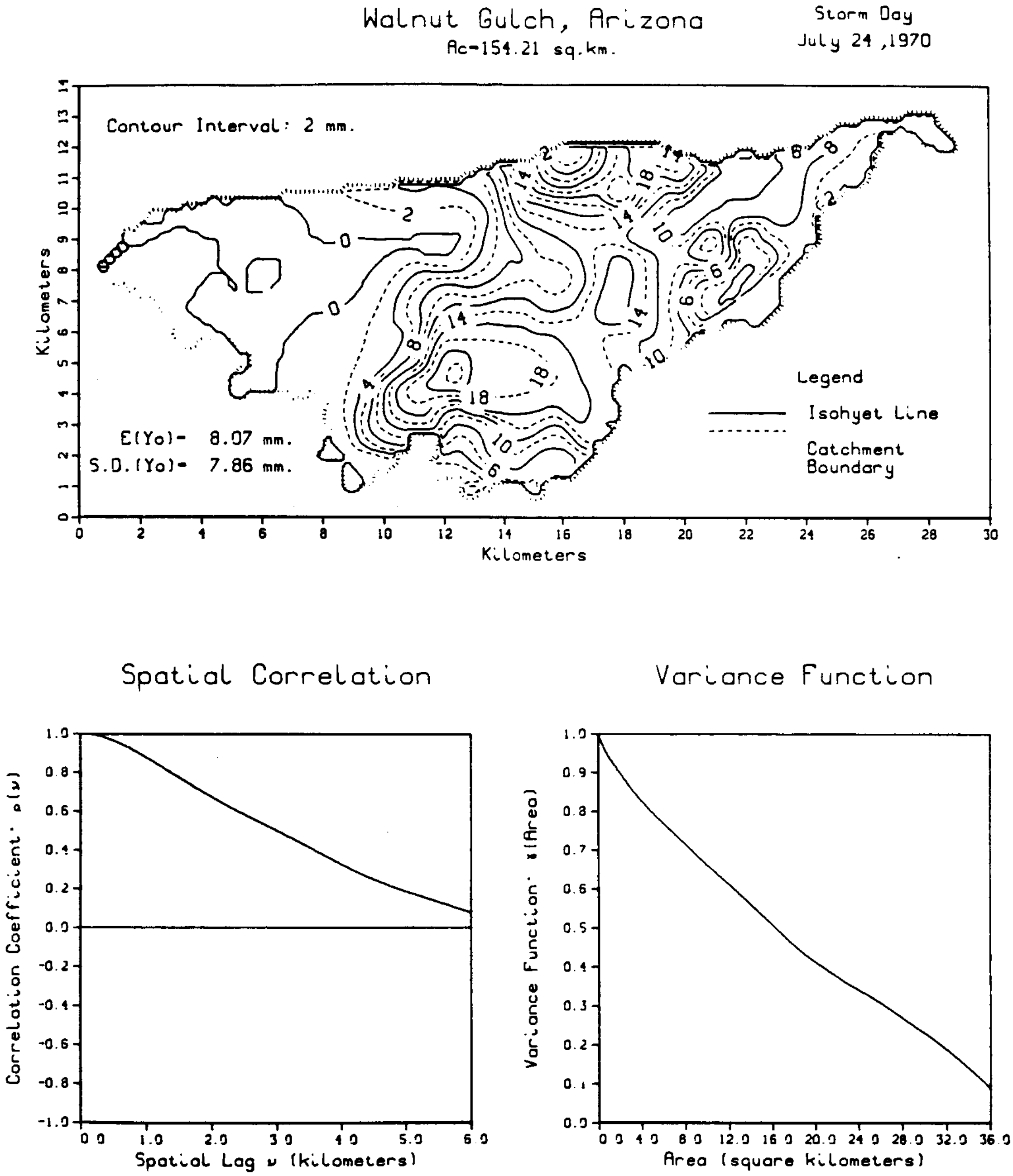

Fig. 9.1.2 24 July 1970 Storm Day

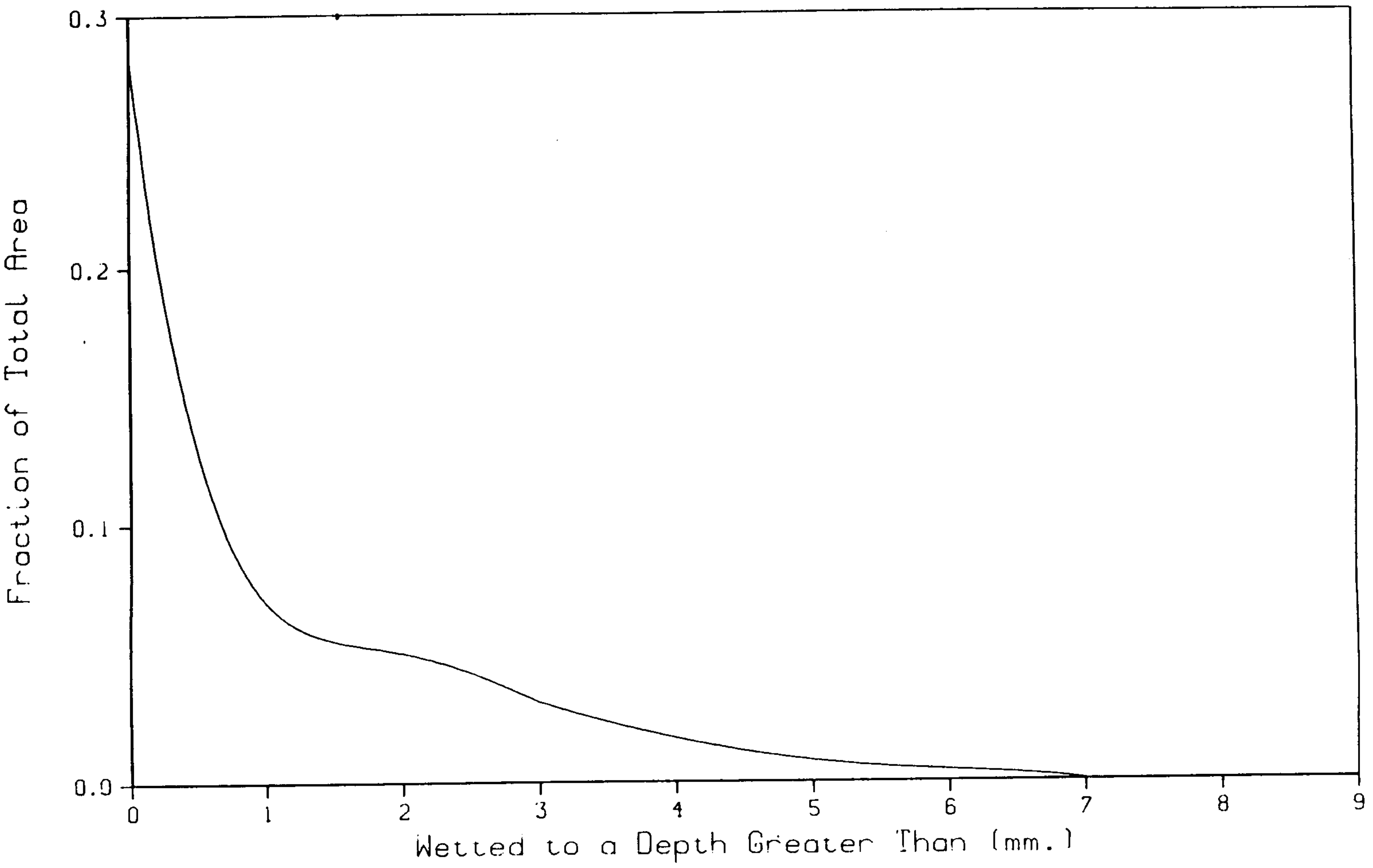

Fig. 9.1.3 22 June 1970 Spatial Distribution of Total Storm Depth

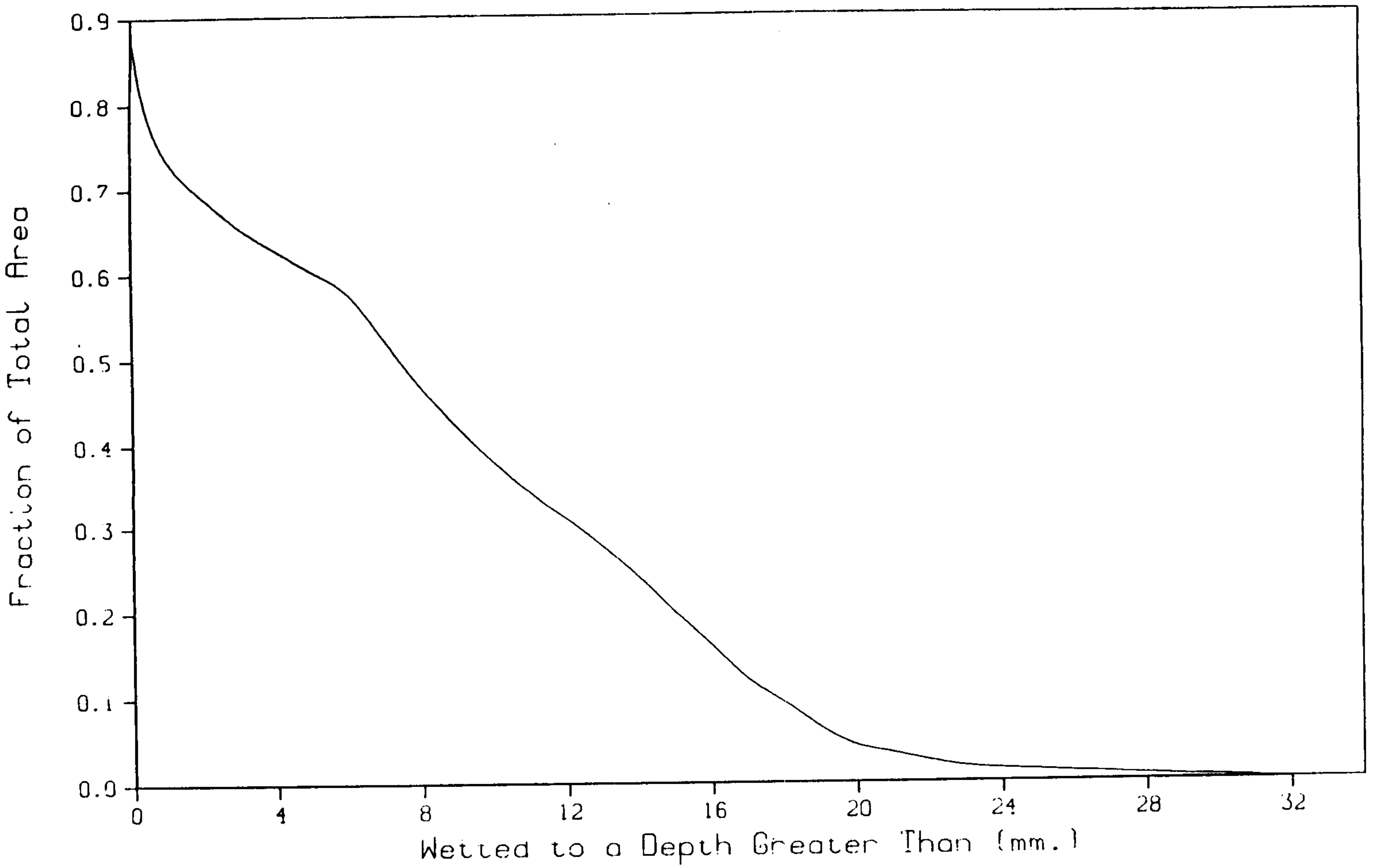

Fig. 9.1.4 24 July 1970 Spatial Distribution of Total Storm Depth

Table 9.1.1

Storm Day June 22 1970

Dry Fraction of Total Basin Area: (Acd/Ac) =0.714

Wetted Fraction of Total Basin Area: (Acw/Ac) =0.286

Expected Value of Point Depth (mm.): E(Y) = 0.270

Variance of Point Depth (mm. sq.): Var(Y) = 0.771

Coef. of Skewness of Point Depth: S.C.(Y) = 4.386

Spatial Distribution of Total Storm Depth		Spatial Correlation		Variance Function	
y (mm.)	Acw/Ac (Y≥y)	v (km.)	rho (v)	A (km.sq.)	Gamma (A)
1	0.070	0.0	1.000	0.00	1.000
2	0.050	0.2	0.976	0.04	0.958
3	0.031	0.4	0.917	0.16	0.896
4	0.017	0.6	0.835	0.36	0.818
5	0.008	0.8	0.740	0.64	0.734
6	0.004	1.0	0.643	1.00	0.654
7	0.000	1.2	0.551	1.44	0.580
8	0.000	1.4	0.470	1.96	0.513
		1.6	0.402	2.56	0.452
		1.8	0.348	3.24	0.396
		2.0	0.306	4.00	0.344
		2.2	0.268	4.84	0.293
		2.4	0.231	5.76	0.244
		2.6	0.192	6.76	0.198
		2.8	0.153	7.84	0.156
		3.0	0.116	9.00	0.120
		3.2	0.081	10.24	0.093
		3.4	0.052	11.56	0.071
		3.6	0.027	12.96	0.055
		3.8	0.011	14.44	0.040
		4.0	0.001	16.00	0.024
		4.2	-.005	17.64	0.010
		4.4	-.008	19.36	0.006
		4.6	-.012	21.16	0.004
		4.8	-.019	23.04	0.004
		5.0	-.029	25.00	0.003
		5.2	-.038	27.04	0.002
		5.4	-.046	29.16	0.001
		5.6	-.054	31.36	0.001
		5.8	-.066	33.64	0.000
		6.0	-.085	36.00	0.000

Table 9.1.2

Storm Day July 24 1970

Dry Fraction of Total Basin Area: (Acd/Ac)=0.122

Wetted Fraction of Total Basin Area: (Acw/Ac)=0.878

Expected Value of Point Depth (mm.): E(Y)= 7.969

Variance of Point Depth (mm. sq.): Var(Y)= 48.831

Coef. of Skewness of Point Depth: S.C.(Y)= 0.549

Spatial Distribution of Total Storm Depth		Spatial Correlation		Variance Function	
y (mm.)	Acw/Ac (Y≥y)	v (km.)	rho (v)	A (km.sq.)	Gamma (A)
1	0.736	0.0	1.000	0.00	1.000
2	0.690	0.2	0.993	0.04	0.997
3	0.654	0.4	0.975	0.16	0.988
4	0.627	0.6	0.948	0.36	0.975
5	0.602	0.8	0.914	0.64	0.957
6	0.575	1.0	0.876	1.00	0.939
7	0.521	1.2	0.835	1.44	0.920
8	0.466	1.4	0.793	1.96	0.899
9	0.420	1.6	0.752	2.56	0.875
10	0.378	1.8	0.712	3.24	0.850
11	0.343	2.0	0.673	4.00	0.825
12	0.309	2.2	0.635	4.84	0.800
13	0.275	2.4	0.600	5.76	0.774
14	0.239	2.6	0.566	6.76	0.747
15	0.198	2.8	0.533	7.84	0.718
16	0.159	3.0	0.499	9.00	0.687
17	0.119	3.2	0.466	10.24	0.655
18	0.092	3.4	0.431	11.56	0.622
19	0.063	3.6	0.396	12.96	0.586
20	0.042	3.8	0.360	14.44	0.548
21	0.033	4.0	0.325	16.00	0.507
22	0.024	4.2	0.292	17.64	0.463
23	0.017	4.4	0.261	19.36	0.423
24	0.014	4.6	0.234	21.16	0.389
25	0.012	4.8	0.209	23.04	0.354
26	0.010	5.0	0.185	25.00	0.323
27	0.008	5.2	0.164	27.04	0.286
28	0.006	5.4	0.142	29.16	0.245
29	0.004	5.6	0.121	31.36	0.201
30	0.003	5.8	0.099	33.64	0.149
31	0.001	6.0	0.077	36.00	0.088
32	0.000				
33	0.000				

CHAPTER 10

Data Archive of Storm Day Spatial Analyses

10.1 The Storm Day Data Archive

The storm day data archive is a series of eight data files which contain the computation results of the processing of each of the 428 storm days. Each file contains the results from processing all the storm days from a single summer rainy season. This archive is invaluable with respect to model parameterization and verification.

Each of the eight files which comprise the data archive contains the series of storm day records for a single summer season. These records are a string of the compacted data tables created by TABLE.FORTRAN.

A computer program entitled FUTURE.FORTRAN is developed to retrieve information from the archive. The listing of FUTURE.FORTRAN is in Appendix I. This program is a foundation on which the analyst can build. Because an analysis of the data may assume many forms, there is no "correct" way to use the computer code. Because of this, in its present form, the program actually does nothing but provide the analyst with the appropriately formatted read statements. The program must be modified by the analyst according to the type of analysis desired.

Because of the size of the archive and the limitations imposed by the computer during this study, large common block variables are used by FUTURE.FORTRAN which total roughly 240 kilobytes of memory. As a safety measure however, all common block variables are preset to zero by the program. A single years' data archive must be placed in a single file.

The 1970 archive is entitled arc70.data, the 1971 archive is entitled arc71.data,..., and the 1977 archive is entitled arc77.data. Arc70.data must be placed in file50, arc71.data placed in file51, and so forth. Following modification of the program according to individual needs, the execution of FUTURE.FORTRAN is ready to begin.

CHAPTER 11

Computational and Ancillary Library Requirements

11.1 The Honeywell Multics Operating System

All data preparation and storm day processing was completed using M.I.T.'s Honeywell mainframe computer. Its operating system is Multics. This machine is somwhat limited in virtual memory space in comparison with more modern computers. The total virtual memory available to the user is only 60 K bytes for variables declared in ordinary "dimension" declaration statements. This limit may be exceeded by using "common" block variable statements. The upper limit using common blocks is 245 K bytes. When using variables stored in common blocks, the user must be certain to preset these variables equal to zero. This is not done by the machine following the completion of a single execution. This could prove disasterous during an analysis, because the variable values are not equal to zero by default at the start of the next execution.

Data files may be placed in any "device" number ranging from 1 through 99. Data cannot be stored in File06 or File07 as with any computer, but in the case of the Multics system, neither File41 nor File42 may be used for data storage.

In comparison with other computer systems, the computational speed of the Multics machine is about 10% faster than the Digital Equipment Corp. MicroVax II mini-computer, and roughly one fourth the speed of the IBM system 370 CMS mainframe.

One peculiarity of the Multics system is that all computer code must be written in lower case lettering as opposed to all upper case (caps). As a consequence, all the computer programs listed in Appendix I are written in lower case. The obvious advantage to this system is that numerical characters are physically taller than alphabetical characters and so confusion is minimized.

11.2 Numerical Libraries

Two different numerical algorithm libraries were used during the course of this study. The reason for using these libraries is that they have been optimized by their authors and are extensively debugged. Because they are proprietary however, these are "black box" routines, which is disadvantageous. Another disadvantage is that the user who wishes to reproduce the results presented in this aspect of the study is unable to should his computer not have access to these libraries.

The primary library accessed to by several of the programs is the I.M.S.L. or International Mathematics and Statistical Library. IMSL is probably the most commonly supported numerical algorithm library and is presently available for use on even mini-computers such as the DEC MicroVax II. The second library accessed to is the N.A.G. or the Numerical Algorithm Group. During storm day processing, we use only one of the IMSL library routines, the bivariate surface interpolator described in Chapter 6.

11.3 Graphics

The sole graphics software used in this study was developed by ISSCO Corporation of San Diego, California. The graphics package is called DISSPLA. DISSPLA is a very robust graphics package and suits the multiple needs of this study. Several federal agencies use DISSPLA, among them, The U.S. Army Corps of Engineers and the U.S.G.S.

DISSPLA has been supported on the Honeywell Mainframe at MIT for only about a year. The version is 9.0 and is not the latest available from ISSCO at this time. Each of the computer codes with graphics routines in them is written with the DISSPLA version 9.0 in mind. These programs will successfully execute on systems which support the current 9.2 version.

As was mentioned in Chapter 9, the execution of the storm day isohyet plot requires the most c.p.u. time to complete. Furthermore, the program requires "restarting" when DISSPLA begins to draw the watershed basin boundary outine. The package will occasionally plot spurious contours off into space or across the face of the isohyet plot. The package also appears to enter an infinite loop if the isohyet coutour interval is too small on some storm days. There is no warning about this in the way of messages or prompts from the software. We have found that if any of the programs which plot isohyets remain in the graphics routine portion of the program for more than three quarters of an hour, execution should be stopped and the contour interval increased.

DISSPLA requires several files during execution and no data should be stored in these files. On the MIT Honeywell Multics, these are file08, file21, file23, file31, file32, file33, file94, and file95.

11.4 User Links to the Numerical Algorithm and Graphics Libraries

The easiest way to link to the numerical algorithm and graphical libraries is to include them in the user "start-up" exec command file. The start up exec is unique to each particular operating system. By invoking global link statements in the start up exec, connection with the libraries is complete all the time and the user need not write specific link statements in the program. For these reasons, there are no links written in any of the programs presented in Chapter 12. The user must write them into a start up executive file instead. It cannot be overemphasized that many of the programs can not be successfully executed unless these links are made.

CHAPTER 12

The Computer Codes

12.1 Introduction

This chapter presents a brief summary of details about each of the computer codes used in this aspect of the study. The hard copies of each program are presented in Appendix I. The purpose of this chapter is not to provide complete documentation for each program but rather to comment about input and output file handling and links to the numerical algorithm and graphics libraries.

All of the computer programs developed are written in the FORTRAN V (Fortran 77) computer language. The programs were developed with other users in mind, and we feel that a small sacrifice in computational efficiency is well worth increased comprehension for the unfamiliar user. For this reason, all programs presented here are self-contained and external functions and subroutines have been avoided. Therefore, the user need not search for the algorithm behind the subroutine call statement.

12.2 TAPEREAD.FORTRAN

The purpose of TAPEREAD.FORTRAN is to read the data from the original data tape that was placed in a file. The code skips over all the information pertaining to rainfall intensity.

In its present form, the program uses a single year's worth of data as an input file of the user's choice. The total number of records in this file must be known before execution begins, as the program will prompt the user for it. The program then writes its results to an output file of the user's choice and will prompt for it.

There are no links to any of the numerical or graphics libraries during the execution of this program.

12.3 SUMMER.FORTRAN

The purpose of SUMMER.FORTRAN is to retain only those station records which belong to the summer rainy season. The program reads in a single year's worth of data from the output file created by TAPEREAD.FORTRAN. The program prompts the user for this input file number and the number of records in the file. The program also prompts the user for the output file number. During execution, the number of records written to the output file are totaled and this value is displayed when the run is complete.

The program does not require links to any of the numerical or graphics libraries during its execution.

12.4 DAY.FORTRAN

The purpose of DAY.FORTRAN is to total the rainfall which fell during each 24 hour storm day at each raingage station. The program results in a newly created data set.

In its present form, DAY.FORTRAN uses the output file from SUMMER.FORTRAN as its input file. The program prompts the user for this input file number and the number of records in the file. The program also prompts the user .for an output file number. During execution, the number of records written to the output file are totaled and this value is displayed when the run is complete.

The program does not require links to any of the numerical or graphics libraries during its execution.

12.5 BALANCE.FORTRAN

The purpose of BALANCE.FORTRAN is to determine the number of storm days in the basin during the course of one summer season, and then to create new records of zero depth for those gages which recorded no rainfall during a storm day.

The program reads in the individual summer season output files created by DAY.FORTRAN as input files. The program prompts the user for this input file number and the number of records in the file. The program also prompts the user for the output file number. During execution, the number of records written to the output file is totaled and this value is displayed when the run is complete.

The BALANCE.FORTRAN program consists of numerous sophisticated logic statements as does DAY.FORTRAN. For three of the eight summer seasons, the program fails to complete execution. This occurs only near the end of the rainy season during the creation of new records for the last raingage listed in the input file. This flaw was never resolved but proves to be no problem in reality. The program will prompt the user with the line number of the input file where failure occurred and similarly, the line number of the output file record. The user must inspect the next to last gage records in the output file and review the dates of the final storm days. These should be either noted or transferred with the editor to complete the final raingage's rainy season record. Using the editor, or some other means, the user must tally the total number of records for later processing.

The program does not require links to any of the numerical or graphics libraries during the its execution.

12.6 COORD.FORTRAN

The purpose of COORD.FORTRAN is to assign spatial cordinates to each station record. The coordinate system is based on a 100 meter ruled mesh (fine resolution) on a watershed basin map provided by the ARS. The coordinate system is unique to this study. The program also filters out any of the station records which do not belong to the final set of 93 storm day gages. The program reads in the individual summer season output files created by BALANCE.FORTRAN as input files. The program prompts the user for this input file number and the number of records in the file. The program also prompts the user for the output file number. During execution, the number of records written to the output file are totaled and this value is displayed when the run is complete.

The program does not require links to any of the numerical or graphics libraries during the its execution.

The eight separate output files created by eight separate executions of COORD.FORTRAN are available on tape from the Director of the Parsons Laboratory at M.I.T. These data files are called: bal70.data which corresponds to the 1970 summer season data file; bal71.data which corresponds to the 1971 season, and so forth.

12.7 STRMSORT.FORTRAN

The purpose of STRMSORT.FORTRAN is to collect the station observations for a single stormday from among all the station records in a single summer rainy season. The 93 station observations are written to output file84 to be used for further processing.

In order to successfully execute this program, the eight summer season output files from COORD.FORTRAN (bal70.data,bal71.data,...,bal77.data) must be placed in file70, file71,...,file77 respectively. The program prompts the user for the last two digits of the year of the rainy season of interest, and the storm day number.

The program does not require links to any of the numerical or graphics libraries during its execution.

12.8 GAGCORR.FORTRAN

The purpose of GAGCORR.FORTRAN is to determine the spatial correlation among the 93 station observations for a single storm day. The program uses file84 as the input file (the results of one exection of STRMSORT.FORTRAN). The output file is written to a file of the user's choice.

The execution of this program requires a link with the DISSPLA graphics package, but not to either of the numerical algorithm libraries.

12.9 ALLCORR.FORTRAN

The purpose of the ALLCORR.FORTRAN program is to determine the spatial correlation of total rainfall depth within the Walnut Gulch coarse mesh random field. The technique used is the radial sweep technique.

The execution of the program requires file84 as an input file (the results of one execution of STRMSORT.FORTRAN). The output file is selected by the user. The user should be cautioned however, because the execution of this program can be expensive. The program requires about 30 minutes of c.p.u. time on the Honeywell Multics mainframe.

This program requires a link with both the DISSPLA and IMSL libraries.

12.10 STORMDAY.FORTRAN

The purpose of the STORMDAY.FORTRAN program is to perform all of the coarse mesh random field analysis described in Chapter 7 and plot the results. The program requires file84 as an input file (results from a single storm day execution of STRMSORT.FORTRAN). The numerical results are written to file91. A single plot file containing the plot of the total depth isohyet map, the spatial correlation plot and the variance function plot are created by the program.

The execution of STORMDAY.FORTRAN requires links to both the IMSL numerical algorithm library and the DISSPLA graphics library.

12.11 STORMWET.FORTRAN

The purpose of STORMWET.FORTRAN is to sample the fine mesh Walnut Gulch random field as described in Chapter 8. The program requires file84 as the input file (the results from one execution of STRMSORT.FORTRAN for the storm day of interest). The numerical results are written to file93 as the output file.

Execution of this program requires a link with the IMSL numerical algorithmn library.

12.12 TABLE.FORTRAN

The purpose of TABLE.FORTRAN is to create two data tables for each storm day once processing is finished. The program requires file91 and file93 (the output files from the execution of STORMDAY.FORTRAN and STORMWET.FORTRAN, respectively) as input files. Two separate output files are created by the program. The first is written to file95 which contains a nicely formatted data table, and the second is written to file02 which contains a compressed data table.

Execution of this program requires no links with either the numerical algorithm or graphics libraries.

12.13 FUTURE.FORTRAN

The purpose of the FUTURE.FORTRAN program is to provide the analyst with the properly formatted statements required to successfully read the eight storm day archive files. Depending on the needs of the particular analysis, the 1970 archive file must be placed into file50, the 1971 archive file must be placed into file51, and so forth. The program must be modified by the analyst depending on what is required because in its present form, the program does nothing, and no output files are as yet designated.

The execution of this program does not require any links with the numerical or graphics libraries.

REFERENCES

Akima, H. A., A Method of Bivariate Interpolation and Smooth Surface Fitting for Irregularly Distributed Data Points, ACM Trans. Math Software 4(2), 148-159, June, 1978.

Battan, L. J., Fundamentals of Meteorology, Prentice-Hall, Englewood Cliffs, NJ, 1984.

Benjamin, J. R. and C. Allin Cornell, Probability, Statistics and Decision for Civil Engineers, McGraw-Hill, New York, NY, 1970.

Bras, R. L. and I. Rodriguez-Iturbe, Random Functions and Hydrology, Addison-Wesley, Reading, MA, 1985.

Cole, F. W., Introduction to Meteorology, Wiley, New York, NY, 1980.

Eagleson, P. S. and Q. Wang, Moments of Catchment Storm Area, Water Resources Research, 21(8), 1185-1194, 1985.

Fennessey, N. M., P. S. Eagleson, Q. Wang and I. Rodriguez-Iturbe, Spatial Characteristics of Observed Precipitation Fields: A Catalog of Summer Storms in Arizona, Volumes I and II, Ralph M. Parsons Laboratory Report No. 307, MIT Department of Civil Engineering, 1986.

Fletcher, J. E., Characteristics of Precipitation (in the Rangelands) of the Southwest, Joint ARS - SCS Hydrology Workshop, New Orleans, LA, 1960.

Koster, R. D., J. Jouzel, R. Souzzo, G. Russel, W. Broecker, D. Rind and P. S. Eagleson, Global Sources of Local Precipitation as Determined by the NASA/GISS GCM, Geophysical Research Letters, 13(1), 121-124, 1986.

Mejia, J. M. and I. Rodriquez-Iturbe, Multidimensional Characterization of the Rainfall Process, Ralph M. Parsons Laboratory Report No. 177, MIT Department of Civil Engineering, November, 1973.

Orlanski, I., A Rational Subdivision of Scales For Atmospheric Processes, Bull. Am. Met. Soc., 56(5), 527-530, 1975.

Osborn, H. B., Quantifiable Differences Between Air-Mass and Frontal-Convective Thunderstorm Rainfall in the Southwestern United States, Proc. Int. Sympos. on Rainfall-Runoff Modeling, May 18-21, Mississippi State University, 1981.

Osborn, H. B., Point to Area Convective Rainfall Simulation, presented at the 13th. American Meteorological Society Conference on Agricultural and Forest Meteorlogy, Purdue University, April 1977.

Osborn, H. B., Precipitation from Thunderstorms in the Southwest, Proc. of the Fifth Conference on Severe Local Storms, American Meteorological Society, St. Louis, MO, October, 1967.

Osborn, H. B., K. G. Renard and J. R. Simanton, Dense Networks to Measure Convective Rainfall in the Southwestern United States, Water Resources Research, 15(6), 1701-1711, 1979.

Osborn, H. B., L. J. Lane and R. S. Kagan, Stochastic Models of Spatial and Temporal Distribution of Thunderstorm Rainfall, Misc. Pub. No. 1275, U.S. Department of Agriculture, 1974.

Osborn, H. B. and E. M. Laursen, Thunderstorm Runoff in Southeastern Arizona, Proc. A.S.C.E., J. Hydraulic Division, 99(HY7), 1129-1145, 1973.

Osborn, H. B. and L. J. Lane, Depth-Area Relationships for Thunderstorm Rainfall in Southeastern Arizona, Trans. of the ASAE, 15(4), 670-680, 1972.

Osborn, H. B. and R. B. Hickok, Variability of Rainfall Affecting Runoff from a Semiarid Rangeland Watershed, Water Resources Research, 4(1), 199-203, 1968.

Osborn, H. B. and W. N. Reynolds, Convective Storm Patterns in the Southwestern United States, Bull. Int. Assoc. Sci. Hydrol. 8(3), 71-83, 1963.

Renard, K. G. and D. L. Brakensiek, Precipitation on Intermountain Rangeland in the Western United States, Proc. of the Fifth Workshop of the United States/Australia Rangelands Panel, Boise, Idaho, June 15-22, 1975.

Restrepo-Posada, P. and P. S. Eagleson, Identification of Independent Rainstorms, Journal of Hydrology 55, 303-319, 1982.

Rodriguez-Iturbe, I. and J. M. Mejia, The Design of Networks in Space and Time, Water Resources Research 10(4), 713-735, 1974.

Rodriguez-Iturbe, I., D. R. Cox and P. S. Eagleson, Spatial Modelling of Total Storm Rainfall, Proc. R. Soc. Lond. A, 403, 27-50, 1986.

Vanmarcke, Erik, Random Fields, Analysis and Synthesis, MIT Press, Cambridge, MA, 1983.

APPENDIX I

Computer Program Listings

```
c                     TAPEREAD.FORTRAN
c
c     The program reads the data tape provided this study by the ARS of the
c     U.S.D.A.
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ............................................................
c
c                  definition of variables
c
c            input variables
c
c     iwat      watershed i.d. number
c     igage     raingage number
c     imonth    month
c     iday      day
c     iyr       year
c     isthr     storm starting hour
c     istmin    storm starting minute
c     idur      storm duration
c     catch     total storm precipitation
c
c           rainfall intensity variables
c
c     ibkpt     break point : number of series of rainfall intensity
c                      records associated with a particular station record
c     ielptm    elasped time since the beginning of the shower (minutes)
c     itmest    time estimate code (0 or blank: not estimated, 1: estimated)
c     acmrnfl   accumulated rainfall (mm. to the nearest tenth)
c     idpest    depth estimate code (0 or blank: not estimated, 1: estimated)
c     rainints  rainfall intensity (mm. per hour)
c     intscd    intensity code (0: not estimated, 1: estimated,
c                       2: not computed)
c
c     ................................................................
c
      %global ansi77
c
      print,'enter the file number the tape dump was written to'
      input,n1
      print,'enter the number of records in file',n1
      input,n2
      print,'enter the file number to be written to'
      input,n4
c
c     read the tape dump records from the file of choice
c
      do 10 ii=1,n2
      read(n1,200)iwat,igage,iyr,imonth,iday,isthr,istmin,idur,catch,ibkpt
          bkpt=ibkpt
          skip=((bkpt/4)+.9)
          iskip=skip
c
c     skip over the data related to rainfall intensity
c
      do 20 j=1,iskip
      read(n1,300)ielptm,itmest,acmrnfl,idpest,rainints,intscd
```

```
20 continue
c
c     write the results to the file of choice
c
      write(n4,100)iwat,igage,iyr,imonth,iday,isthr,istmin,idur,catch
        k=k+1
          ii=iskip+ii
          bkpt=0
          skip=0
          iskip=0
10    continue
100   format(i2,i3,1x,3i2,1x,2i2,1x,i5,f6.2)
200   format(i2,i3,1x,3i2,1x,2i2,1x,i4,f6.2,i3)
300   format(4(i4,i1,f6.2,i1,f7.2,i1))
c
      print,'there were ',k,' records written to file',n4
c
      end
```

```
c                          SUMMER.FORTRAN
c
c     This program will take files created by TAPEREAD.FORTRAN
c     and eliminate all station observation shower records which
c     occur in months other than those which belong to the
c     summer rainy season which begins at noon June 1 and
c     ends at noon October 1
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     .............................................................
c
c                    definition of variables
c
c            input variables
c
c     iwat      watershed i.d. number
c     igage     raingage number
c     imonth    month
c     iday      day
c     iyr       year
c     isthr     storm starting hour
c     istmin    storm starting minute
c     idur      storm duration (minutes)
c     catch     total storm precipitation (mm.)
c
c           output variables
c
c     iwat      watershed i.d. number
c     igag      raingage number
c     imo       month
c     idy       day
c     iyear     year
c     isthur    storm starting hour
c     istrtmin  storm starting minute
c     idurat    storm duration (minutes)
c     rain      total storm precipitation (mm.)
c
c     ...............................................................
c
      %global ansi77
      dimension   igage(7000),imonth(7000),iday(7000)
      dimension   iyr(7000),isthr(7000),istmin(7000)
      dimension   idur(7000),catch(7000)
c
c
      print,'enter the master file number to be summer sorted'
      input,n2
      print,'enter the number of records in file',n2
      input,n1
      print,'enter the file number to be written to'
      input,n5
c
      print,'enter last 2 digits of the year of interest'
      input,n6
c
c
c     this section reads in the walnut gulch data for comparative
```

```
c      purposes
c
       do 10 ii=1,n1
       read(n2,200) iwat,igage(ii),iyr(ii),imonth(ii),iday(ii),isthr(ii),
      &istmin(ii),idur(ii),catch(ii)
10     continue
c
200    format(i2,i3,1x,3i2,1x,2i2,1x,i5,f6.2)
c
       do 20 ii=1,n1
           a=imonth(ii)
        if((a.ge.6).and.(a.le.9).and.(iyr(ii).eq.n6)) then
           igag=igage(ii)
           iyear=iyr(ii)
           imo=imonth(ii)
           idy=iday(ii)
           isthur=isthr(ii)
           istrtmin=istmin(ii)
           idurat=idur(ii)
           rain=catch(ii)
c
        write(n5,200) iwat,igag,iyear,imo,idy,isthur,istrtmin,idurat,rain
c
           k=k+1
c
c       October 1 st. shower
c
        else if((a.eq.10).and.(iday(ii).eq.1).and.(iyr(ii).eq.n6).and.
      &(isthr(ii).lt.12)) then
c
        write(n5,200) iwat,igag,iyear,imo,idy,isthur,istrtmin,idurat,rain
           k=k+1
        else
           go to 20
        end if
20      continue
300     continue
        print,'there were ',k,' records written to file ',n5
        end
```

```
c                           DAY.FORTRAN
c
c     The program creates new station records for each 24 hour storm day
c     that occurred during the summer rainy season at Walnut Gulch, Az.
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     .............................................................
c
c                  definition of variables
c
c             input variables
c
c      iwat      watershed i.d. number
c      igage     raingage number
c      imonth    month
c      iday      day
c      iyr       year
c      isthr     shower starting hour
c      istmin    shower starting minute
c      idur      shower duration
c      catch     total shower precipitation
c
c       program variables
c
c      isthur    shower starting hour
c      istmn     shower starting minute
c      idurat    shower duration
c      duratmin  shower duration in minutes
c      durathr   shower duration in hours
c      strtmin   shower starting minute
c      frctstmn  shower starting minute (fraction: strtmin/60.0)
c      strthr    shower starting hour
c      totstrhr  total shower starting time (hour + fraction of an hour)
c      endhr     shower ending time
c
c       created data variables: all these variable are subscripted _____(i)
c      iwa       watershed i.d. number
c      igag      raingage number
c      imo       month
c      idy       day
c      iyear     year
c      dystrthr  elapsed time in hours that the day begins since Jan.1,1970
c      dyendhr   elapsed time in hours that the day ends since Jan.1,1970
c      pptysday  rainfall which lasted beyond noon time of today (yesterday's
c                    leftover precipitation from the final shower
c      ppttoday  total precipitation recorded between noon today and noon
c                    tomorrow
c      pptints   average shower intensity (shower depth / shower duration)
c      pptomoro  rainfail which will last beyond noon today into tomorrow
c                    (equals pptysday)
c      totrain   total stormday rainfall (mm.)
c      rain      zero (0.0) mm. total rainfall depth
c
c      j         created day counter
c      i         created variable array subscript
c      ii        raw variable array subscript
c
```

```
c     ...............................................................
c
      %global ansi77
c
      common /var1 /iwat(12000)
      common /var2 /igage(12000)
      common /var3 /imonth(12000)
      common /var3 /iday(12000)
      common /var4 /iyr(12000)
      common /var5 /isthr(12000)
      common /var6 /istmin(12000)
      common /var8 /date(12000)
      common /var9 /idur(12000)
      common /var10 /catch(12000)
      common /var11 /rain(12000)
      common /var12 /totrain(12000)
      common /var13 /iwa(12000)
      common /var14 /igag(12000)
      common /var15 /imo(12000)
      common /var16 /idy(12000)
      common /var17 /iyear(12000)
      common /var18 /durathr(12000)
      common /var19 /totstrhr(12000)
      common /var20 /endhr(12000)
c
      print,'enter file number to be split into 24 hour blocks'
      input,n1
      print,'enter the number of records in file',n1
      input,n2
      print,'enter final file number to be written to'
      input,n4
c
c
c     *************************************************************************
c
c     preset the common block variables to zero
c
c     *************************************************************************
c
c
      do 50 i=1,12000
      iwat(i)=0.0
      igage(i)=0.0
      imonth(i)=0.0
      iday(i)=0.0
      iyr(i)=0.0
      isthr(i)=0.0
      istmin(i)=0.0
      date(i)=0.0
      idur(i)=0.0
      catch(i)=0.0
      rain(i)=0.0
      totrain(i)=0.0
      iwa(i)=0.0
      igag(i)=0.0
      imo(i)=0.0
      idy(i)=0.0
      iyear(i)=0.0
      durathr(i)=0.0
      totstrhr(i)=0.0
      endhr(i)=0.0
50    continue
c
c
c
c     ************************************************************************
c
c     this section reads in the origional walnut gulch data
c     from the input file
c
c     ************************************************************************
c
c
      do 10 ii=1,n2
      read(n1,100)iwat(ii),igage(ii),iyear(ii),imonth(ii),iday(ii),isthr(ii),
     &istmin(ii),idur(ii),catch(ii)
100   format(i2,i3,1x,3i2,1x,2i2,1x,i5,f6.2)
          duramin=idur(ii)
          durathr(ii)=duramin/60.0
          strtmin=istmin(ii)
          frctstmn=strtmin/60.0
          strthr=isthr(ii)
          totstrhr(ii)=strthr+frctstmn
          endhr(ii)=totstrhr(ii)+durathr(ii)
c
10    continue
c
c     ************************************************************************
c
c      initialize the counters
c
c     ************************************************************************
c
          j=0
c
      do 30 ii=1,n2
c
          j=j+1
c
c     ************************************************************************
c
c      this section creates partial shower records; those days in which
c      a shower begins before noon time and extends on into the next day
c
c     ************************************************************************
c
      if((totstrhr(ii).lt.12.).and.
     &(endhr(ii).ge.12)) then
c
          pptints=catch(ii)/durathr(ii)
          ppttoday=((12.0-totstrhr(ii))*pptints)
          pptomoro=((endhr(ii)-12.0)*pptints)
          pptysday=pptomoro
c
c     ************************************************************************
c
c      Check to see if the partial shower begins on the last day of the
c      month and continues on into the first day of the next month
c
c     ************************************************************************
```

```
c
c                         JUNE-JULY
c
      if((imonth(ii).eq.7).and.(iday(ii).eq.1))  then
          imo(j)=6
          idy(j)=30
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          j=j+1
          imo(j)=7
          idy(j)=1
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=pptysday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
c                         JULY-AUGUST
c
      else if((imonth(ii).eq.8).and.(iday(ii).eq.1))  then
          imo(j)=7
          idy(j)=31
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          j=j+1
          imo(j)=8
          idy(j)=1
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=pptysday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
c                       AUGUST-SEPTEMBER
c
      else if((imonth(ii).eq.9).and.(iday(ii).eq.1))  then
          imo(j)=8
          idy(j)=31
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          j=j+1
          imo(j)=9
          idy(j)=1
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=pptysday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
c                   SEPTEMBER-OCTOBER
c
      else if((imonth(ii).eq.10).and.(iday(ii).eq.1)) then
          imo(j)=9
          idy(j)=30
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
      end if
c
          idy(j)=iday(ii)-1
          imo(j)=imonth(ii)
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          j=j+1
          idy(j)=iday(ii)
          imo(j)=imonth(ii)
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=pptysday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
      end if
c
c     **********************************************************************
c
c     Adress those showers which begin and end before noon of the
c     calendar day's record, and don't occur on the first day of the month
c
c     **********************************************************************
c
      if((totstrhr(ii).lt.12.).and.
     &(endhr(ii).lt.12).and.
     &(iday(ii).ne.1)) then
          idy(j)=iday(ii)-1
          imo(j)=imonth(ii)
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=catch(ii)
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
```

```
            go to 20
         end if
c
c        **********************************************************************
c
c        Adress those  showers which begin and end before noon
c        of the day's record, but do occur on the first day of the month
c
c        **********************************************************************
c
         if((totstrhr(ii).lt.12.).and.
        &(endhr(ii).lt.12).and.
        &(iday(ii).eq.1)) then
c
c        **********************************************************************
c
c         Check to see if the morning shower begins on the first day of the
c         month and, therefore belongs to the last day of the prior month
c
c        **********************************************************************
c
c                                JUNE-JULY
c
          if((imonth(ii).eq.7).and.(iday(ii).eq.1)) then
              imo(j)=6
              idy(j)=30
              iwa(j)=iwat(ii)
              igag(j)=igage(ii)
              iyr(j)=iyear(ii)
              rain(j)=catch(ii)
              date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
              go to 20
c
c                             JULY-AUGUST
c
          else if((imonth(ii).eq.8).and.(iday(ii).eq.1)) then
              imo(j)=7
              idy(j)=31
              iwa(j)=iwat(ii)
              igag(j)=igage(ii)
              iyr(j)=iyear(ii)
              rain(j)=catch(ii)
              date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
              go to 20
c
c                           AUGUST-SEPTEMBER
c
          else if((imonth(ii).eq.9).and.(iday(ii).eq.1)) then
              imo(j)=8
              idy(j)=31
              iwa(j)=iwat(ii)
              igag(j)=igage(ii)
              iyr(j)=iyear(ii)
              rain(j)=catch(ii)
              date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
              go to 20
c
```

```
c                         SEPTEMBER-OCTOBER
c
      else if((imonth(ii).eq.10).and.(iday(ii).eq.1)) then
          imo(j)=9
          idy(j)=30
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=catch(ii)
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
      end if
c
          go to 20
c
      end if
c
c     ************************************************************************
c
c     Adress those showers which begin and end during the storm day:
c     after noon time today and before noon time tomorrow
c
c     ************************************************************************
c
      if((totstrhr(ii).ge.12.).and.
     &(endhr(ii).lt.36.)) then
          idy(j)=iday(ii)
          imo(j)=imonth(ii)
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=catch(ii)
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
      end if
c
c     ************************************************************************
c
c     Adress the long duration  showers which begin and complete after day's
c     end: after noon time today and after noon time tomorrow
c
c     ************************************************************************
c
      if((totstrhr(ii).ge.12.).and.
     &(endhr(ii).ge.36)) then
          pptints=catch(ii)/durathr(ii)
          ppttoday=((36.0-totstrhr(ii))*pptints)
          pptomoro=((endhr(ii)-36.0)*pptints)
          pptysday=pptomoro
c
c     ************************************************************************
c
c      Check to see if the partial shower begins on the last day of the
c      month and continues on into the first day of the next month
c
c     ************************************************************************
c
```

```
c                         JUNE-JULY
c
      if((imonth(ii).eq.6).and.(iday(ii).eq.30)) then
         imo(j)=6
         idy(j)=30
         iwa(j)=iwat(ii)
         igag(j)=igage(ii)
         iyr(j)=iyear(ii)
         rain(j)=ppttoday
         date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
         j=j+1
         imo(j)=7
         idy(j)=1
         iwa(j)=iwat(ii)
         igag(j)=igage(ii)
         iyr(j)=iyear(ii)
         rain(j)=pptysday
         date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
         go to 20
c
c                    JULY-AUGUST
c
      else if((imonth(ii).eq.7).and.(iday(ii).eq.31)) then
         imo(j)=7
         idy(j)=31
         iwa(j)=iwat(ii)
         igag(j)=igage(ii)
         iyr(j)=iyear(ii)
         rain(j)=ppttoday
         date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
         j=j+1
         imo(j)=8
         idy(j)=1
         iwa(j)=iwat(ii)
         igag(j)=igage(ii)
         iyr(j)=iyear(ii)
         rain(j)=pptysday
         date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
         go to 20
c
c                      AUGUST-SEPTEMBER
c
      else if((imonth(ii).eq.8).and.(iday(ii).eq.31)) then
         imo(j)=8
         idy(j)=31
         iwa(j)=iwat(ii)
         igag(j)=igage(ii)
         iyr(j)=iyear(ii)
         rain(j)=ppttoday
         date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
         j=j+1
         imo(j)=9
         idy(j)=1
         iwa(j)=iwat(ii)
         igag(j)=igage(ii)
```

```
          iyr(j)=iyear(ii)
          rain(j)=pptysday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
c                     SEPTEMBER-OCTOBER
c
      else if((imonth(ii).eq.10).and.(iday(ii).eq.1)) then
          imo(j)=9
          idy(j)=30
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
c
      end if
c
          idy(j)=iday(ii)
          imo(j)=imonth(ii)
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=ppttoday
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          j=j+1
          idy(j)=iday(ii)+1
          imo(j)=imonth(ii)
          iwa(j)=iwat(ii)
          igag(j)=igage(ii)
          iyr(j)=iyear(ii)
          rain(j)=pptomoro
          date(j)=(iyr(j)*10000)+(imo(j)*100)+idy(j)
c
          go to 20
      end if
c
20    continue
c
c
30    continue
c
c     **********************************************************************
c
c     Total the total rainfall for the storm day and write the results
c     to the file of choice
c
c     **********************************************************************
c
          i=j
          j=0

      do 40 j=1,i
c
c     **********************************************************************
c
```

```
c      there were ten total records created for this day
c
c      ******************************************************************************
c
       if((date(j).eq.date(j+9)).and.
      &(igag(j).eq.igag(j+9))) then
           k=k+1
           totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)+rain(j+4)+rain(j+5)+
      &rain(j+6)+rain(j+7)+rain(j+8)+rain(j+9)
       write(n4,200) iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
           j=j+9
c
c      ******************************************************************************
c
c      there were nine total records created for this day
c
c      ******************************************************************************
c
       else if((date(j).eq.date(j+8)).and.
      &(igag(j).eq.igag(j+8))) then
           k=k+1
           totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)+rain(j+4)+rain(j+5)+
      &rain(j+6)+rain(j+7)+rain(j+8)
       write(n4,200) iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
           j=j+8
c
c      ******************************************************************************
c
c      there were eight total records created for this day
c
c      ******************************************************************************
c
       else if((date(j).eq.date(j+7)).and.
      &(igag(j).eq.igag(j+7))) then
           k=k+1
           totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)+rain(j+4)+rain(j+5)+
      &rain(j+6)+rain(j+7)
       write(n4,200) iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
           j=j+7
c
c      ******************************************************************************
c
c      there were seven total records created for this day
c
c      ******************************************************************************
c
       else if((date(j).eq.date(j+6)).and.
      &(igag(j).eq.igag(j+6))) then
           k=k+1
           totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)+rain(j+4)+rain(j+5)+
      &rain(j+6)
       write(n4,200) iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
           j=j+6
c
c      ******************************************************************************
c
c      there were six total records created for this day
c
c      ******************************************************************************
c
```

```
      else if((date(j).eq.date(j+5)).and.
     &(igag(j).eq.igag(j+5))) then
          k=k+1
          totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)+rain(j+4)+rain(j+5)
      write(n4,200)iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
          j=j+5
c
c     ********************************************************************************
c
c     there were five total records created for this day
c
c     ********************************************************************************
c
      else if((date(j).eq.date(j+4)).and.
     &(igag(j).eq.igag(j+4))) then
          k=k+1
          totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)+rain(j+4)
      write(n4,200)iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
          j=j+4
c
c     ********************************************************************************
c
c     there were four total records created for this day
c
c     ********************************************************************************
c
      else if((date(j).eq.date(j+3)).and.
     &(igag(j).eq.igag(j+3))) then
          k=k+1
          totrain(k)=rain(j)+rain(j+1)+rain(j+2)+rain(j+3)
      write(n4,200)iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
          j=j+3
c
c     ********************************************************************************
c
c     there were three total records created for this day
c
c     ********************************************************************************
c
      else if((date(j).eq.date(j+2)).and.
     &(igag(j).eq.igag(j+2))) then
          k=k+1
          totrain(k)=rain(j)+rain(j+1)+rain(j+2)
      write(n4,200)iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
          j=j+2
c
c     ********************************************************************************
c
c     there were two total records created for this day
c
c     ********************************************************************************
c
      else if((date(j).eq.date(j+1)).and.
     &(igag(j).eq.igag(j+1))) then
          k=k+1
          totrain(k)=rain(j)+rain(j+1)
      write(n4,200)iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
          j=j+1
c
c     ********************************************************************************
```

```
c
c     there was one total record created for this day
c
c     ************************************************************************
c
      else
          k=k+1
          totrain(k)=rain(j)
      write(n4,200)iwa(j),igag(j),iyr(j),imo(j),idy(j),totrain(k)
c         j=j+1
c
      end if
c
40    continue
c
200   format(i2,i3,1x,3i2,f6.2)
c
500   continue
      print,'you are done, please check file',n4
      print,'file',n4,' has ',k,' records in it'
      end
```

```
c                        BALANCE.FORTRAN
c
c     The program  creates new station records for each of the raingages
c     which received no rainfall during each 24 hour storm day created by
c     DAY.FORTRAN  that occurred during the summer rainy season at
c     Walnut Gulch, Az.
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ..............................................................
c
c                  definition of variables
c
c                 input variables
c
c      iwat      watershed i.d. number
c      igage     raingage number
c      imonth    month
c      iday      day
c      iyr       year
c      isthr     shower starting hour
c      istmin    shower starting minute
c      idur      shower duration
c      catch     total shower precipitation
c
c              output variables
c
c      iwa       watershed i.d. number
c      igag      raingage number
c      imo       month
c      idy       day
c      iyear     year
c      iiidate   date of the storm day
c      iidate    date of the storm day
c      zd        storm day rainfall (mm.)
c      rain      zero rainfall
c
c      ..............................................................
c
c
       dimension   iwat(6200),igage(6200),iyear(6200),imonth(6200)
       dimension   iday(6200),zd(6200),iiidate(6200),iidate(6200),idate(100)
c
       print,'enter the file number to be balanced'
       input,n3
       print100,n3
       input,n2
       print,'enter the file number to which the results will be written'
       input,n1
100    format('enter the number of records in file',i4)
c
c      *************************************************************************
c
c      the section reads in the storm data from input file to be balanced
c
c      *************************************************************************
c
       do 140 i=1,n2
```

```
      read(n3,1000) iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
140   continue
c
1000  format(i2,i3,1x,3i2,f6.2)
2000  format(i2,i3,1x,i6,f6.2)
c
c     **********************************************************************
c
c     examine all the dates in the unbalanced records and compile a list
c     of all the different dates for which there were storm days
c
c     **********************************************************************
c
          j=0
          i=0
      do 10 i=1,n2
          j=j+1
          iiidate(j)=(iyear(i)*10000)+(imonth(i)*100)+iday(i)
          iidate(i)=(iyear(i)*10000)+(imonth(i)*100)+iday(i)
      do 20 k=1,j-1
          diff=abs(iiidate(j)-iiidate(k))
      if((diff.eq.0.0).and.(j.gt.1)) then
          j=j-1
          go to 10
      end if
20    continue
10    continue
c
c     **********************************************************************
c
c     sort the storm day dates from the earliest to latest
c
c     **********************************************************************
c
      do 40 i=1,j
      do 50 k=1,j-1
          a=iiidate(k)
          b=iiidate(k+1)
      if(a.lt.b) then
          idate(k)=a
          idate(k+1)=b
      else
          iiidate(k)=b
          iiidate(k+1)=a
          idate(k)=b
          idate(k+1)=a
      end if
50    continue
40    continue
      do 55 k=1,j
      print45,idate(k)
55    continue
45    format('storm date idate(k)=',i8)
c
c     **********************************************************************
c
c     begin to balance the records now
c
c     **********************************************************************
c
```

```
c     initialize the counting variables
c
         k=1
         i=1
         rain=0.0
30    continue
c
c     ************************************************************************
c
c     examine the date counter
c
c     ************************************************************************
c
      if((k.eq.j+1).and.(i.lt.n2)) then
         k=1
c
c     ************************************************************************
c
c     there are one or more null record dates before the first storm date
c     first gage first record
c
c     ************************************************************************
c
      else if ((idate(k).lt.iidate(i)).and.
     &(i.eq.1)) then
      write(n1,2000) iwat(i),igage(i),idate(k),rain
         k=k+1
         go to 30
c
c     ************************************************************************
c
c     there are one or more null record date before the first storm date
c     for gages other than the first gage
c
c     ************************************************************************
c
      else if ((idate(k).lt.iidate(i)).and.
     &(igage(i).ne.igage(i-1))) then
      write(n1,2000) iwat(i),igage(i),idate(k),rain
         k=k+1
         go to 30
c
c     ************************************************************************
c
c     there is a null record date before the storm date at this gage
c     for all gauges
c
c     ************************************************************************
c
      else if ((idate(k).lt.iidate(i)).and.
     &(igage(i).eq.igage(i+1)).and.
     &((i+1).le.n2).and.(k.gt.1).and.(k.lt.j)) then
      write(n1,2000) iwat(i),igage(i),idate(k),rain
         k=k+1
         go to 30
c
c     ************************************************************************
c
c     there is a null record date before the last storm date at this gage
c     for all gauges
```

```
c
c      ************************************************************************
c
       else if  ((idate(k).lt.iidate(i)).and.
     &(igage(i).ne.igage(i+1)).and.
     &((i+1).lt.n2).and.(k.gt.1).and.(k.lt.j)) then
       write(n1,2000) iwat(i),igage(i),idate(k),rain
           k=k+1
           go to 30
c
c      ************************************************************************
c
c      there is rain on the same day as the day of the date list
c
c      ************************************************************************
c
       else if  ((idate(k).eq.iidate(i)).and.
     &(igage(i).eq.igage(i+1)).and.
     &((i+1).le.n2).and.(k.lt.j)) then
       write(n1,1000) iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
           i=i+1
           k=k+1
           go to 30
c
c      ************************************************************************
c
c      on the last record of the same gauge with still null records to
c      write except for the last gauge
c
c      ************************************************************************
c
       else if  ((idate(k).eq.iidate(i)).and.
     &(igage(i).ne.igage(i+1)).and.
     &((i+1).lt.n2).and.(k.lt.j)) then
       write(n1,1000) iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
           k=k+1
           go to 30
c
c      ************************************************************************
c
c      on the last record of the same gauge with still null records to
c      write except for the last gauge
c
c      ************************************************************************
c
       else if  ((idate(k).gt.iidate(i)).and.
     &(igage(i).ne.igage(i+1)).and.
     &((i+1).lt.n2).and.(k.le.j)) then
       write(n1,2000) iwat(i),igage(i),idate(k),rain
       if(k.eq.j) then
           k=1
           i=i+1
       else
           k=k+1

       end if
           go to 30
c
c      ************************************************************************
c
```

```
c     storm on the last record of the gauge except for the last gauge
c
c     **********************************************************************
c
      else if  ((idate(k).eq.iidate(i)).and.
     &(igage(i).ne.igage(i+1)).and.
     &((i+1).lt.n2).and.(k.eq.j)) then
      write(n1,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
          k=1
          i=i+1
          go to 30
c
c     **********************************************************************
c
c     on the last record of the file with still null records to write
c
c     **********************************************************************
c
      else if  ((idate(k).eq.iidate(i)).and.
     &(i.eq.n2).and.(k.lt.j)) then
      write(n1,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
          k=k+1
          go to 30
c
c     **********************************************************************
c
c     on the last record of the file with still null records to write
c     before the final last record storm date
c
c     **********************************************************************
c
      else if  ((idate(k).lt.iidate(i)).and.
     &(i.eq.n2).and.(k.lt.j)) then
      write(n1,2000)iwat(i),igage(i),idate(k),rain
          k=k+1
          go to 30
c
c     **********************************************************************
c
c     there is rain on the same day as the day of the date list, last
c     gauge, last record
c
c     **********************************************************************
c
      else if  ((idate(k).eq.iidate(i)).and.
     &(i.eq.n2).and.(k.lt.j)) then
      write(n1,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
          k=k+1
          go to 30
c
c     **********************************************************************
c
c     on the last record of the file with still null records to write
c
c     **********************************************************************
c
      else if  ((idate(k).gt.iidate(i)).and.
     &(i.eq.n2).and.(k.lt.j)) then
      write(n1,2000)iwat(i),igage(i),idate(k),rain
      if(k.eq.j) then
```

```
          go to 5
      else
          k=k+1
          go to 30
      end if
c
c     ************************************************************************
c
c     on the last storm on the last record of the file with nothing more
c     to write
c
c     ************************************************************************
c
      else if  ((idate(k).eq.iidate(i)).and.
     &(i.eq.n2).and.(k.eq.j)) then
      write(n1,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i)
c
          go to 5
      else
      print,'there is an alternative in the logic somewhere'
      print,' the algorithmn failed at input file record number ',i
      print,' the algorithmn failed at output file record number ',k
      print,'the input record date is',iidate(i)
      print,'the output record date is',idate(k)
      print,'iidate(i)=',iidate(i),' idate(k)=',idate(k)
          print,'n2=',n2,' i=',i,' k=',k,' j=',j
      print,' this algorithmn will only fail during the creation of the'
      print,'final raingage created records.'
      print,'edit the output file by "hand" by checking on the next to'
      print,'last output file gage records for the storm day dates and'
      print,'complete the final raingage storm day records'
c
      end if
c
5     continue
      print,' there are ',j,' storm days this year '
      end
c
```

```
c                          COORD.FORTRAN
c
c     This program will assign spatial corrdinates to each of the
c     93 storm day raingage records for each storm day created by
c     BALANCE.FORTRAN.  All other station observation records will
c     be skipped
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ....................................................
c
c                  definition of variables
c
c           input variables
c
c     iwat       watershed i.d. number
c     igage      raingage number
c     imonth     month
c     iday       day
c     iyr        year
c     catch      total storm precipitation (mm.)
c
c           output variables
c
c     iwat       watershed i.d. number
c     igage      raingage number
c     imonth     month
c     iday       day
c     iyr        year
c     catch      total storm precipitation (mm.)
c     x          spatial coordinate
c     y          spatial coordinate
c
c     ............................................................
c
      %global ansi77
      dimension   iwat(7100),igage(7100),imonth(7100),iday(7100)
      dimension   iyr(7100),catch(7100)
c
c
      print,'enter file number file to be read in and given coordinates'
      input,n3
      print,'enter the number of records in file',n3
      input,n2
      print,'enter file number file to be written to'
      input,n1
c
c     *********************************************************************
c
c     this section reads in the walnut gulch data
c
c     *********************************************************************
c
      do 10 ii=1,n2
      read(n3,200) iwat(ii),igage(ii),iyr(ii),imonth(ii),iday(ii),catch(ii)
10    continue
c
c     *********************************************************************
```

```
c
c     assign spatial corrdinates to each record depending on the gage number
c
c     ************************************************************************
c
      do 20 i=1,n2
          j=igage(i)
      if(j.eq.1) then
          x=10
          y=82
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.2) then
          x=22
          y=94
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.3) then
          x=21
          y=72
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.4) then
          x=39
          y=100
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.5) then
          x=36
          y=81
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.7) then
          x=52
          y=102
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.8) then
          x=48
          y=88
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.9) then
          x=56
          y=67
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.10) then
          x=52
          y=58
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.11) then
          x=70
          y=101
      write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
     &x,y
      else if(j.eq.12) then
          x=63
          y=89
```

```
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.13) then
     x=69
     y=77
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.14) then
     x=64
     y=45
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.15) then
     x=87
     y=105
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.16) then
     x=79
     y=90
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.17) then
     x=91
     y=74
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.18) then
     x=77
     y=54
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.19) then
     x=88
     y=41
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.20) then
     x=85
     y=23
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.21) then
     x=101
     y=107
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.22) then
     x=94
     y=91
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.23) then
     x=104
     y=87
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.24) then
     x=96
     y=62
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.25) then
     x=108
     y=41
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.26) then
     x=98
     y=28
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.27) then
     x=116
     y=93
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.28) then
     x=117
     y=73
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.29) then
     x=129
     y=59
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.30) then
     x=126
     y=33
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.31) then
     x=134
     y=113
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.32) then
     x=136
     y=91
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.33) then
     x=129
     y=78
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.34) then
     x=120
     y=48
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.35) then
     x=128
     y=39
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.36) then
     x=122
     y=17
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.37) then
     x=144
     y=33
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.38) then
     x=157
     y=111
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.39) then
     x=146
     y=86
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.40) then
     x=144
     y=77
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.41) then
     x=144
     y=53
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.42) then
     x=138
     y=22
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.43) then
     x=161
     y=120
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.44) then
     x=161
     y=90
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.45) then
     x=158
     y=76
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.46) then
     x=163
     y=60
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.47) then
     x=154
     y=43
 write(n1,100)iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.48) then
     x=167
     y=29
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.49) then
      x=156
      y=25
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.50) then
      x=178
      y=124
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.51) then
      x=167
      y=103
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.52) then
      x=178
      y=93
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.53) then
      x=172
      y=68
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.54) then
      x=196
      y=114
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.55) then
      x=213
      y=112
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.56) then
      x=195
      y=97
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.57) then
      x=176
      y=84
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.58) then
      x=186
      y=68
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.59) then
      x=195
      y=57
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.60) then
      x=210
      y=98
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.61) then
     x=207
     y=89
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.62) then
     x=217
     y=83
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.63) then
     x=208
     y=71
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.65) then
     x=227
     y=100
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.66) then
     x=234
     y=87
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.67) then
     x=238
     y=119
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.68) then
     x=245
     y=105
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.69) then
     x=252
     y=127
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.70) then
     x=257
     y=119
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.71) then
     x=130
     y=100
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.72) then
     x=191
     y=84
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.73) then
     x=150
     y=125
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.74)  then
     x=117
     y=114
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.75)  then
     x=73
     y=115
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.76)  then
     x=37
     y=69
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.77)  then
     x=144
     y=8
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.78)  then
     x=169
     y=17
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.79)  then
     x=73
     y=65
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.80)  then
     x=120
     y=86
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.81)  then
     x=104
     y=55
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.82)  then
     x=214
     y=90
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.83)  then
     x=107
     y=97
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.87)  then
     x=146
     y=101
 write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
&x,y
 else if(j.eq.88)  then
     x=187
     y=109
```

```
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.89) then
           x=176
           y=115
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.90) then
           x=177
           y=105
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.91) then
           x=205
           y=106
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.384) then
           x=106
           y=94
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.385) then
           x=151
           y=88
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.386) then
           x=104
           y=97
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.512) then
           x=59
           y=89
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.537) then
           x=144
           y=32
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.560) then
           x=210
           y=99
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else if(j.eq.587) then
           x=144
           y=103
       write(n1,100) iwat(i),igage(i),iyr(i),imonth(i),iday(i),catch(i),
      &x,y
       else
           jj=jj+1
       end if
20     continue
           jjj=n2-jj
c
100    format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
200    format(i2,i3,1x,3i2,f6.2)
```

```
300   format('there were',i6,' records written to file',i4)
      print300,jjj,n1
c
      end
```

```
c                     STRMSORT.FORTRAN
c
c     This program gathers together the 93 station observation records
c     for a single storm day.
c     The files called bal70.data, bal71.data ... and ... bal77.data
c     must be placed in file70, file71, file73 ... file77.
c     (file digits correspond to the last two digits of
c     the year in question).
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ..........................................................
c
c                  definition of variables
c
c            input and output variables
c
c     iwat      watershed i.d. number
c     igage     raingage number
c     imonth    month
c     iday      day
c     iyr       year
c     isthr     storm starting hour
c     istmin    storm starting minute
c     idur      storm duration
c     catch    total storm precipitation
c
c     n3        file number (corresponds to the year)
c     n1        storm day number
c     n2        number of storm days in year n3 (file n3)
c     n4        number of records in file n3
c
c     ...............................................................
c
      %global ansi77
      dimension   iwat(7000),igage(7000),imonth(7000),iday(7000)
      dimension   iyr(7000),catch(7000),x(7000),y(7000)
c
c     *************************************************************************
c
c      select file file corresponding to the year of interest
c
c     *************************************************************************
c
      print,'enter the last two digits of the year of interest'
      input,n3
c
c     *************************************************************************
c
c     select the storm day number for the storm day of interest
c
c     *************************************************************************
c
c
      print,'enter storm number desired'
      input,n1
c
c     *************************************************************************
```

```
c
c     determine the number of storm days this summer season
c
c     **************************************************************************
c
      if(n3.eq.70) then
         n2=49
      else if(n3.eq.71) then
         n2=52
      else if(n3.eq.72) then
         n2=58
      else if(n3.eq.73) then
         n2=45
      else if(n3.eq.74) then
         n2=63
      else if(n3.eq.75) then
         n2=48
      else if(n3.eq.76) then
         n2=53
      else if(n3.eq.77) then
         n2=63 .
      end if
c
c     **************************************************************************
c
c     determine the number of records in this summer season's file
c
c     **************************************************************************
c
      if(n3.eq.70) then
         n4=4557
      else if(n3.eq.71) then
         n4=4836
      else if(n3.eq.72) then
         n4=5394
      else if(n3.eq.73) then
         n4=4185
      else if(n3.eq.74) then
         n4=5859
      else if(n3.eq.75) then
         n4=4464
      else if(n3.eq.76) then
         n4=4929
      else if(n3.eq.77) then
         n4=5859
      end if
c
c
      print,'do you wish to continue with this run, enter a 1 for'
      print,'a go , and a 2 for a stop'
      input,fa
      if (fa.eq.1.0) then
         go to 400
      else
         go to 600
      end if
c
c     **************************************************************************
c
c     this section reads in the walnut gulch data
```

```
c
c      ********************************************************************
c
400    continue
c
       do 10 i=1,n4
       read(n3,200)iwat(i),igage(i),iyr(i),imonth(i),iday(i),
      &catch(i),x(i),y(i)
10     continue
       rewind(n3)
c
c      ********************************************************************
c
c      this section writes the 93 station observations for this
c      particular storm day to file84
c
c      ********************************************************************
c
       do 30 i=n1,n4,n2
       write(84,200)iwat(i),igage(i),iyr(i),imonth(i),iday(i),
      &catch(i),x(i),y(i)
30     continue
c
200    format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
600    continue
c
       end
```

```
c                         GAGECORR.FORTRAN
c
c     The program determines the spatial correlation among the 93
c     storm day station observation total rainfall depth records
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ...........................................................
c
c                  definition of variables
c
c                     input variables
c
c     iwat      watershed i.d. number
c     igage     raingage number
c     imonth    month
c     iday      day
c     iyr       year
c     zd        total station storm day rainfall  (mm.)
c     xd        station spatial coordinate
c     yd        station spatial coordinate
c
c                 random field variables
c
c     zi        node total depth  (mm.)
c     xi        node spatial coordinate  (0.1 km.)
c     yi        node spatial coordinate  (0.1 km.)
c
c                       Moments
c
c     avedepth  average depth of the random field: E(Y)  (mm.)
c     vardepth  variance of the depth of the random field: VAR(Y)  (sq. mm.)
c
c                  Spatial correlation
c
c     x         lag distance  (km.)
c     y         spatial correlation of gage depth at a particular lag distance
c     ipairs    number of node pairs created at this lag distance
c
c     ...........................................................
c
c
      dimension   iwat(93),igage(93),iyear(93),imonth(93)
      dimension   iday(93),zd(93),xd(93),yd(93)
      dimension   ipairs(0:200),gagdst(0:7000),check(0:7000)
      dimension   y(101),x(101),y2(101),x2(101)
      dimension   t1(0:500),b1(0:500),b2(0:500)
      dimension   avedepth(93),amonth(6)
      dimension   ipairs2(30),y3(30),x3(30)
c
      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
c
      print,'Are you linked to the DISSPLA ?'
c
         nd=93
c
c     ***************************************************************************
```

```
c
c     the section reads in the storm day data from file84
c
c     ****************************************************************************
c
      do 140 i=1,nd
      read(84,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
     &xd(i),yd(i)
140   continue
c
          iiday=iday(1)
          iiyear=iyear(1)+1900
          l=imonth(1)-5
          amnth=amonth(1)

c
      print,'...........................  '
      print900,amnth,iiday,iiyear
      print,'...........................  '
c
900   format('storm day  ',a4,i3,i5)
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
c
c     ****************************************************************************
c
c     calculate the average gauge depth: E(Yo)
c
c     ****************************************************************************
c
      do 240 i=1,nd
          a1=zd(i)+a1
          d1=d1+1
240   continue
c
          avedpth=a1/d1
c
c     ****************************************************************************
c
c     this portion will calculate the variance of the gauge depth:
c     VAR(Yo)
c
c     ****************************************************************************
c
      do 160 j=1,nd
          l=l+1
          a2=((zd(j)-avedpth)**2.)+a2
160   continue
          vardepth=a2/(l-1)
      print,' '
      print 360,avedpth
      print 350,vardepth
360   format('the average gauge depth=',f10.3,' mm.'/)
350   format('the variance of the gauge depth=',f10.3,' mm squared'/)
c
c     ****************************************************************************
c
c     compare the distance with the object gauge and ech of the other
c     gauges, then see if a particular pair been already been examined
c
c     ****************************************************************************
```

```
c
          j=0
          l=0
          i=0
      do 20 l=1,nd
      do 30 i=1,nd
          j=j+1
          gagdst(j)=sqrt(((xd(l)-xd(i))**2.)+((yd(l)-yd(i))**2.))
          check(j)=gagdst(j)*(igage(l)*igage(i))+(igage(l)+igage(i))
c
c     **********************************************************************
c
c     gagdst equals the distance between the gage in question and then
c     object gauge.  Aside from itself (needed for lag zero), check to see
c     if we have compared these two before, or if the lag distance is
c     greater than 100 (10 km.)
c
c     **********************************************************************
c
          c1=xd(l)-xd(i)
          c2=yd(l)-yd(i)
      if(((c1.eq.0.0).and.(c2.eq.0.0)).or.(j.eq.1)) then
          go to 35
      else if(gagdst(j).gt.100) then
          j=j-1
          go to 30
      else
          k=0
      do 40 k=1,j-1
          diff=abs(check(k)-check(j))
      if(diff.eq.0.0) then
          j=j-1
          go to 30
      end if
40    continue
      end if
c
c     **********************************************************************
c
c     round the lag distance up or down to the nearest integer value
c
c     **********************************************************************
c
35    continue
c
          igagdst=gagdst(j)
          a3=gagdst(j)-igagdst
      if(a3.ge.0.5) then
          lag=igagdst+1
      else
          lag=igagdst
      end if
c
c     **********************************************************************
c
c     calculate the components which comprise the correlation for this
c     particular lag.
c
c
c     **********************************************************************
```

```
c
          t1(lag)=((zd(i)-avedpth)*(zd(l)-avedpth))+t1(lag)
          b1(lag)=((zd(i)-avedpth)**2.)+b1(lag)
          b2(lag)=((zd(l)-avedpth)**2.)+b2(lag)
          ipairs(lag)=ipairs(lag)+1
c
30    continue
20    continue
c
c
c     **********************************************************************
c
c     calculate the correlation coefficient for each lag up to a lag
c     of ten kilometers.  Check to see if the variance of either gauge
c     equals zero.
c
c
c     **********************************************************************
c
      do 60 i=0,100
      if((b1(i).eq.0.0).or.(b2(i).eq.0.0)) then
          f1=i
          y2(i+1)=0.0
          x2(i+1)=f1/10.
          go to 60
      else if(ipairs(i).eq.1) then
          ipairs(i)=0
          f1=i
          y2(i+1)=0.0
          x2(i+1)=f1/10.
          go to 60
      else
          k3=k3+1
          y(i+1)=t1(i)/(sqrt(b1(i)*b2(i)))
          f1=i
          x(i+1)=f1/10.
          y2(i+1)=0.0
          x2(i+1)=f1/10.
      end if
60    continue
c
c     *******************************************************************
c
c     smooth the above point values using a moving weighted average
c
c     *******************************************************************
c
c     set the smoothed curve lag zero equal to 1.0
c
c     *******************************************************************
c
          k4=1
          x3(k4)=0.0
          y3(k4)=1.0
c
      do 180 i=8,101,5
          c1=0.
          d1=0.
          g1=0.
      do 190 j=i,i+4
```

```
           g1=j+g1
           c1=ipairs(j)+c1
           d1=ipairs(j)*y(j+1)+d1
190     continue
           k4=k4+1
           x3(k4)=g1/50.
           y3(k4)=d1/c1
           ipairs2(k4)=c1
180     continue
c
c       **********************************************************************
c
c       this section of the program starts the disspla plotting routines
c
c       **********************************************************************
c
        print,'entering the plotting routines'
        call comprs
        call page(11.0,8.5)
        call basalf('stand')
        call mx4alf('greek',1h*)
        call yaxang(0.0)
        call xname('Spatial Lag *n)  (kilometers)$',100)
        call yname('Correlation Coefficient: *r(n))$',100)
        call area2d(7.0,5.0)
        call headin('Intergauge Spatial Correlation$',100,1.5,1)
        call thkfrm(0.01)
        call messag('Storm Day$',9,5.2,5.45)
        call messag(amnth,4,5.0,5.2)
        call messag(',$',1,5.9,5.2)
        call intno(iiday,5.6,5.2)
        call intno(iiyear,6.0,5.2)
        call frame
        call graf(0.,1.,10.,-1.,0.2,1.0)
        call marker(15)
        call poly3
        call curve(x3,y3,k4,1)
        call thkcrv(2)
        call curve(x2,y2,101,0)
        call endpl(0)
        call donepl
        end
```

```
c                            ALLCORR.FORTRAN
c
c
c     The program samples the coarse mesh random field and determines
c     the spatial correlation by the radial sweep algorithmn
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ............................................................
c
c                   definition of variables
c
c                      input variables
c
c     iwat      watershed i.d. number
c     igage     raingage number
c     imonth    month
c     iday      day
c     iyr       year
c     zd        total station storm day rainfall (mm.)
c     xd        station spatial coordinate
c     yd        station spatial coordinate
c
c                  random field variables
c
c     zi        node total depth (mm.)
c     xi        node spatial coordinate (0.1 km.)
c     yi        node spatial coordinate (0.1 km.)
c
c                      Moments
c
c     avedepth  average depth of the random field: E(Y) (mm.)
c     vardepth  variance of the depth of the random field: VAR(Y) (sq. mm.)
c
c                    Spatial correlation
c
c     x         lag distance (km.)
c     y         spatial correlation of the random field at a particular lag
c     pairs     number of node pairs created at this lag distance
c
c     ............................................................
c
      dimension   zi(147,70),amonth(6)
      dimension   iwat(93),igage(93),iyear(93),imonth(93)
      dimension   iday(93),zd(93),xd(93),yd(93)
      dimension   xi(147),yi(70),wk(558),iwk(13173)
      dimension   y(200),x(200),y2(200),x2(200)
      dimension   pairs(0:200)
      dimension   t1(0:200),b1(0:200),b2(0:200)
c
      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
c
      print,'Are you linked to the IMSL and DISSPLA  libraries?'
c
      print,' if you wish to create a plot, enter a 1'
      input,choice
c
```

```
      print,' enter the file number for the results to be written to'
      input,nln
      print,'enter the number of input records, currently 93'
      input,nd
c
c     ************************************************************************
c
c     the section reads in the storm data from file84 and divides
c     these coordinates in two (both x and y) for use in the coarse
c     mesh random field.
c
c     ************************************************************************
c
      do 140 i=1,nd

      read(84,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
     &xd(i),yd(i)
          xd(i)=xd(i)/2.
          yd(i)=yd(i)/2.
140   continue
c
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
c
c
          iiday=iday(1)
          iiyear=1900+iyear(1)
          iii=imonth(1)-5
          amnth=amonth(iii)
      print,'.............................       '
      print1005,amonth(iii),iiday,iiyear
      print,'.............................       '
1005  format('storm day ',a4,i3,i5)
c
c     ************************************************************************
c
c     this portion generates the x and y coordinates of the nodes in
c     coarse mesh matrix: xi and yi
c
c     ************************************************************************
c
      nxi=147
      nyi=70
      izi=147
      nxi1=146
      nyi1=69
      nxi2=nxi*2
      nyi2=nyi*2
      do 120 j=1,nxi
          xi(j)=j
120    continue
      do 130 i=1,nyi
          yi(i)=i
130    continue
c
c     ************************************************************************
c
c     invoke the IMSL bivariate surface interpolator subroutine IQHSCV
c     to generate the coarse total rainfall depth surface
c
c     ************************************************************************
c
```

```
c
      call iqhscv(xd,yd,zd,nd,xi,nxi,yi,nyi,zi,izi,iwk,wk,ier)
c
c     ***********************************************************************
c
c     this section of the program creates the numerical watershed boundary
c     filter.  It assigns  the nodes of the coarse mesh matrix outside of
c     the Walnut Gulch basin random field with depth values of -1. mm.
c     The values generated by bivariate surface interpolator for nodes
c     within the random field are left intact and untouched.
c
c     ***********************************************************************
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
c
          l=j/2
          m=i/2
c
c      These logic statements define the coarse mesh basin boundary.
c
       if((i.eq.2)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.4)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.6)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.8)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
     &(j.le.150)))) then
          go to 25
       else if(i.eq.10) then
          zi(l,m)=-1.
       else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
     &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
          go to 25
       else if(i.eq.12) then
          zi(l,m)=-1.
       else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
     &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
          go to 25
       else if(i.eq.14) then
          zi(l,m)=-1.
       else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
     &(j.le.160)))) then
          go to 25
       else if(i.eq.16) then
          zi(l,m)=-1.
       else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
     &(j.le.162)))) then
          go to 25
       else if(i.eq.18) then
```

```
          zi(1,m)=-1.
      else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
     &(j.le.164)))) then
          go to 25
      else if(i.eq.20) then
          zi(1,m)=-1.
      else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
     &(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
          go to 25
      else if(i.eq.22) then
          zi(1,m)=-1.
      else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
     &(j.le.166)))) then
          go to 25
      else if(i.eq.24) then
          zi(1,m)=-1.
      else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
     &(j.le.166)))) then
          go to 25
      else if(i.eq.26) then
          zi(1,m)=-1.
      else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
     &(j.le.170)))) then
          go to 25
      else if(i.eq.28) then
          zi(1,m)=-1.
      else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
     &(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
          go to 25
      else if(i.eq.30) then
          zi(1,m)=-1.
          go to 25
      else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
          go to 25
      else if(i.eq.32) then
          zi(1,m)=-1.
      else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
          go to 25
      else if(i.eq.34) then
          zi(1,m)=-1.
          go to 25
      else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
          go to 25
      else if(i.eq.36) then
          zi(1,m)=-1.
      else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
          go to 25
      else if(i.eq.38) then
          zi(1,m)=-1.
      else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
          go to 25
      else if(i.eq.40) then
          zi(1,m)=-1.
      else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
     &(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
          go to 25
      else if(i.eq.42) then
          zi(1,m)=-1.
      else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
     &(j.le.172)))) then
```

```
           go to 25
       else if(i.eq.44) then
           zi(1,m)=-1.
       else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
           go to 25
       else if(i.eq.46) then
           zi(1,m)=-1.
       else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
           go to 25
       else if(i.eq.48) then
           zi(1,m)=-1.
       else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
     &(j.le.182)))) then
           go to 25
       else if(i.eq.50) then
           zi(1,m)=-1.
       else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
           go to 25
       else if(i.eq.52) then
           zi(1,m)=-1.
       else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
           go to 25
       else if(i.eq.54) then
           zi(1,m)=-1.
       else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
           go to 25
       else if(i.eq.56) then
           zi(1,m)=-1.
       else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
     &(j.lè.196)))) then
           go to 25
       else if(i.eq.58) then
           zi(1,m)=-1.
       else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
           go to 25
       else if(i.eq.60) then
           zi(1,m)=-1.
       else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
           go to 25
       else if(i.eq.62) then
           zi(1,m)=-1.
       else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
     &(j.le.221)))) then
           go to 25
       else if(i:eq.64) then
           zi(1,m)=-1.
       else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
           go to 25
       else if(i.eq.66) then
           zi(1,m)=-1.
       else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
           go to 25
       else if(i.eq.68) then
           zi(1,m)=-1.
       else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
           go to 25
       else if(i.eq.70) then
           zi(1,m)=-1.
       else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
           go to 25
```

```
else if(i.eq.72) then
    zi(l,m)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 25
else if(i.eq.74) then
    zi(l,m)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 25
else if(i.eq.76) then
    zi(l,m)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 25
else if(i.eq.78) then
    zi(l,m)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 25
else if(i.eq.80) then
    zi(l,m)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 25
else if(i.eq.82) then
    zi(l,m)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 25
else if(i.eq.84) then
    zi(l,m)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 25
else if(i.eq.86) then
    zi(l,m)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.88) then
    zi(l,m)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
    go to 25
else if(i.eq.90) then
    zi(l,m)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 25
else if(i.eq.92) then
    zi(l,m)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
    go to 25
else if(i.eq.94) then
    zi(l,m)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
    go to 25
else if(i.eq.96) then
    zi(l,m)=-1.
else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
    go to 25
else if(i.eq.98) then
    zi(l,m)=-1.
else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
    go to 25
else if(i..eq.100) then
    zi(l,m)=-1.
else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
    go to 25
```

```
 else if(i.eq.102) then
      zi (1,m)=-1.
 else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
      go to 25
 else if(i.eq.104) then
      zi (1,m)=-1.
 else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
      go to 25
 else if(i.eq.106) then
      zi (1,m)=-1.
 else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
      go to 25
 else if(i.eq.108) then
      zi (1,m)=-1.
 else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
      go to 25
 else if(i.eq.110) then
      zi (1,m)=-1.
 else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
      go to 25
 else if(i.eq.112) then
      zi (1,m)=-1.
 else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
&.and.(j.le.256)))) then
      go to 25
 else if(i.eq.114) then
      zi (1,m)=-1.
 else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
&.and.(j.le.257)))) then
      go to 25
 else if(i.eq.116) then
      zi (1,m)=-1.
 else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
&.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
      go to 25
 else if(i.eq.118) then
      zi (1,m)=-1.
 else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
&.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
      go to 25
 else if(i.eq.120) then
      zi (1,m)=-1.
 else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
&.and.(j.le.283)))) then
      go to 25
 else if(i.eq.122) then
      zi (1,m)=-1.
 else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
&.and.(j.le.281)))) then
      go to 25
 else if(i.eq.124) then
      zi (1,m)=-1.
 else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
&.and.(j.le.279)))) then
      go to 25
 else if(i.eq.126) then
      zi (1,m)=-1.
 else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
      go to 25
 else if(i.eq.128) then
```

```
          zi(1,m)=-1.
      else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
         go to 25
      else if(i.eq.130) then
          zi(1,m)=-1.
      else if(i.eq.132) then
          zi(1,m)=-1.
         go to 20
      else if(i.eq.134) then
         zi(1,m)=-1.
         go to 20
      else if(i.eq.136) then
         zi(1,m)=-1.
         go to 20
      else if(i.eq.138) then
         zi(1,m)=-1.
         go to 20
      else if(i.eq.140) then
         zi(1,m)=-1.
         go to 20
      end if
c
25    continue
20    continue
10    continue
c
c     ************************************************************************
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, and imposes the
c     minimum depth filter where any node in the random field with a
c     value of less than or equal to 0.01 mm. is reassigned a value of
c     0.0 mm
c
c     ************************************************************************
c
      do 260 i=1,nxi
      do 270 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
         go to 270
      else if(zi(i,j).le.0.1) then
         zi(i,j)=0.0
      end if
c
270   continue
260   continue
c
c     ************************************************************************
c
c     calculate the average rainfall depth:E(Y) in the random field
c
c     ************************************************************************
c
      do 240 i=1,nxi
      do 250 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
         go to 250
      end if
```

```
          a1=zi(i,j)+a1
          d1=d1+1
250     continue
240     continue
c
          avedepth=a1/d1
c
c       ***************************************************************************
c
c       this portion will calculate the variance of the rainfall depth: VAR(Y)
c       of the random field.
c
c       ***************************************************************************
c
          l=0
c
        do 155 i=1,nxi
        do 160 j=1,nyi
c
        if(zi(i,j).eq.-1.) then
          go to 160
        end if
          l=l+1
          a2=((zi(i,j)-avedepth)**2.)+a2
160     continue
155     continue
          vardepth=a2/(l-1)
        print,' '
        print 360,avedepth
        print,' '
        print 350,vardepth
        print,' '
360     format('the average point depth=',f10.3,' mm.')
350     format('the variance of the point depth=',f10.3,' mm squared')
c
c       ***************************************************************************
c
c       this portion of the program will determine the two dimensional
c       spatial correlation of the random field.  The technique used is
c       a radial sweep sampling technique.   The correlation will be
c       determined for lag distances which range from 0.0 to 6.0 km.
c
c       ***************************************************************************
c
          l=0
c
       do 170 j=1,nyi
       do 180 i=1,nxi
c
       if(zi(i,j).eq.-1.) then
          go to 180
         end if
c
c       ***************************************************************************
c
c       sweep the left hand boundary of the matrix vertically upwards to
c       nyi from the one level above level j of the pivot point zi(i,j)
c       to the boundary point zi(1,k)
c
c       ***************************************************************************
```

```
c
      do 190 k=j+1,nyi
          l=1
c
c     ************************************************************************
c
c     check to see if zi(i,j) is located at the upper left side of the
c     grid
c
c     ************************************************************************
c
      if((i.eq.1).and.(j.eq.nyi)) then
          go to 190
      end if
          deltai=l-i
          deltaj=k-j
c
c     ************************************************************************
c
c     test for the smaller absolute  value of deltai or deltaj
c     and use as a step variable, unless it's equal to zero
c
c     ************************************************************************
c
      if(abs(deltai).eq.0.) then
          ll=abs(deltaj)
      else if(abs(deltaj).eq.0.) then
          ll=abs(deltai)
      else if((abs(deltai).lt.abs(deltaj))) then
          ll=abs(deltai)
      else
          ll=abs(deltaj)
      end if
c
c
c     ************************************************************************
c
c     find the largest common divisor
c
c     ************************************************************************
c
      do 210 k2=1,ll
           ab=abs(k2-1-ll)
           ai=deltai/ab
           iai=deltai/ab
           bi=ai-iai
           aj=deltaj/ab
           jaj=deltaj/ab
           bj=aj-jaj
c
c     ************************************************************************
c
c
c     if both bj and bi equal zero, we have the largest common divisor
c
c     ************************************************************************
c
      if((bi.eq.0.).and.(bj.eq.0.)) then
          go to 220
      end if
```

```
210   continue
220   continue
c
          k3=ab
c
c     **********************************************************************
c
c     if k3 equals 1, then only the boundary point and the pivot point
c     would be included in this determination, which we do not want
c
c     **********************************************************************
c
c      if(k3.eq.1) then
c         go to 190
c      end if
c
c     **********************************************************************
c
c     calculate the increments between the points which lie on the ray
c     in question
c
c     **********************************************************************
c
      do 230 12=0,k3
c
c     **********************************************************************
c
c     ie and je are the x and y coordinates of those points which
c     lie on the ray in question
c
c     **********************************************************************
c
          ie=i+(12*iai)
          je=j+(12*jaj)
c
c     **********************************************************************
c
c     check to see if this point value zi(ie,je) along the ray equals -1.0
c
c     **********************************************************************
c
      if(zi(ie,je).eq.-1.) then
          go to 230
      else
c
c     **********************************************************************
c
c     calculate the lag distance between zi(i,j) and each zi(ie,je) along ray
c     and use this value as a counter for the correlation analysis.  Round up
c     or round down
c
c     **********************************************************************
c
          zz=sqrt(((ie-i)**2.)+((je-j)**2.))
          jj=sqrt(((ie-i)**2.)+((je-j)**2.))
          zzz=zz-jj
      if(zzz.ge.0.5) then
          jj=jj+1
      end if
c
```

```
      if(jj.le.105) then
          t1(jj)=((zi(i,j)-avedepth)*(zi(ie,je)-avedepth))+t1(jj)
          b1(jj)= ((zi(i,j)-avedepth)**2.)+b1(jj)
          b2(jj)=((zi(ie,je)-avedepth)**2.)+b2(jj)
          pairs(jj)=pairs(jj)+1
      end if
c
      end if
230   continue
190   continue
c
c     **************************************************************************
c
c
c     sweep the top of the  boundary of the matrix from left to right to nxi
c     with nyi from the same level as level j of the pivot point zi(i,j)
c     to the boundary point zi(l,k)
c
c     **************************************************************************
c
      do 280 l=2,nxi
          k=nyi
c
c     **************************************************************************
c
c     check to see if zi(i,j) lies on the top row
c
c     **************************************************************************
c
      if(j.eq.nyi) then
          l=i+1
      end if
          deltai=l-i
          deltaj=k-j
c
c     **************************************************************************
c
c     test for the smaller absolute  value of deltai or deltaj
c     and use as a step increment, unless it's equal to zero
c
c     **************************************************************************
c
      if(abs(deltai).eq.0.) then
          l1=abs(deltaj)
      else if(abs(deltaj).eq.0.) then
          l1=abs(deltai)
      else if((abs(deltai).lt.abs(deltaj))) then
          l1=abs(deltai)
      else
          l1=abs(deltaj)
      end if
c
c     **************************************************************************
c
c     find the largest common divisor
c
c     **************************************************************************
c
      do 290 k2=1,l1
          ab=abs(k2-1-l1)
          ai=deltai/ab
          iai=deltai/ab
          bi=ai-iai
          aj=deltaj/ab
          jaj=deltaj/ab
          bj=aj-jaj
c
c     **********************************************************************
c
c     if both bj and bi equal zero, we have the largest common divisor
c
c
c     **********************************************************************
c
      if((bi.eq.0.).and.(bj.eq.0.)) then
          go to 265
      end if
290   continue
265   continue
c
          k3=ab
c
c
c     **********************************************************************
c
c     if k3 equals 1, then only the boundary point and the pivot point
c     would be included in this determination, which we do not want
c
c
c     **********************************************************************
c
c      if(k3.eq.1) then
c         go to 280
c      end if
c
c
c     **********************************************************************
c
c     calculate the increments between the points which lie on the ray
c     in question
c
c
c
c     **********************************************************************
c
      do 300 12=0,k3
c
c
c
c     **********************************************************************
c
c     ie and je are the x and y coordinates of those points which
c     lie on the ray in question
c
c
c
c     **********************************************************************
c
          ie=i+(12*iai)
          je=j+(12*jaj)
```

```
c
c
c
c     **********************************************************************
c
c     check to see if this point value zi(ie,je) along the ray equals -1.0
c
c     **********************************************************************
c
      if(zi(ie,je).eq.-1.) then
          go to 300
      else
c
c     **********************************************************************
c
c     calculate the lag distance between zi(i,j) and each zi(ie,je) along ray
c     and use this value as a counter for the correlation analysis.  Round up
c     or round down
c
c     **********************************************************************
c
          zz=sqrt(((ie-i)**2.)+((je-j)**2.))
          jj=sqrt(((ie-i)**2.)+((je-j)**2.))
          zzz=zz-jj
      if(zzz.ge.0.5) then
          jj=jj+1
      end if
c
      if(jj.le.105) then
          t1(jj)=((zi(i,j)-avedepth)*(zi(ie,je)-avedepth))+t1(jj)
          b1(jj)= ((zi(i,j)-avedepth)**2.)+b1(jj)
          b2(jj)=((zi(ie,je)-avedepth)**2.)+b2(jj)
          pairs(jj)=pairs(jj)+1
      end if
c
      end if
300   continue
280   continue
c
c     **********************************************************************
c
c     sweep the right hand boundary of the matrix vertically downward from
c     nyi to the same  level as level j of the pivot
c     point zi(i,j)  to the boundary point zi(l,k)
c
c     **********************************************************************
c
c     check to see if we are on the top row of the grid, if so neglect
c     sweeping the right side of the grid and instead, advance to a new
c     value of both i and j
c
c     **********************************************************************
c
      if(j.eq.nyi) then
          go to 170
      end if
c
      do 310 k=nyi1,j,-1
          l=nxi
c
```

```
c     ************************************************************************
c
c     check to see if zi(i,j) lies at the upper right corner of the mesh or
c     if zi(i,j) lies along the right hand boundary, if so we are done
c
c     ************************************************************************
c
      if((i.eq.nxi).and.(j.eq.k)) then
          go to 180
      else if(j.eq.nyi) then
          go to 170
      end if
          deltai=l-i
          deltaj=k-j
c
c     ************************************************************************
c
c     test for the smaller absolute  value of deltai or deltaj
c     and use as a stepping increment, unless it's equal to zero
c
c     ************************************************************************
c
      if(abs(deltai).eq.0.) then
          l1=abs(deltaj)
      else if(abs(deltaj).eq.0.) then
          l1=abs(deltai)
      else if((abs(deltai).lt.abs(deltaj))) then
          l1=abs(deltai)
      else
          l1=abs(deltaj)
      end if
c
c     ************************************************************************
c
c     find the largest common divisor
c
c     ************************************************************************
c
      do 320 k2=1,l1
           ab=abs(k2-1-l1)
           ai=deltai/ab
           iai=deltai/ab
           bi=ai-iai
           aj=deltaj/ab
           jaj=deltaj/ab
           bj=aj-jaj
c
c     ************************************************************************
c
c     if both bj and bi equal zero, we have the largest common divisor
c
c     ************************************************************************
c
      if((bi.eq.0.).and.(bj.eq.0.)) then
          go to 325
      end if
320   continue
325   continue
c
          k3=ab
```

```
c
c      ***********************************************************************
c
c      if k3 equals 1, then only the boundary point and the pivot point
c      would be included in this determination, which we do not want
c
c      ***********************************************************************
c
c       if(k3.eq.1) then
c          go to 310
c       end if
c
c      ***********************************************************************
c
c      calculate the increments between the points which lie on the ray
c      in question
c
c      ***********************************************************************
c
       do 330 l2=0,k3
c
c
c      ***********************************************************************
c
c      ie and je are the x and y coordinates of those points which
c      lie on the ray in question
c
c
c      ***********************************************************************
c
           ie=i+(l2*iai)
           je=j+(l2*jaj)
c
c
c      ***********************************************************************
c
c      check to see if this point value along the ray equals -1.0
c
c      ***********************************************************************
c
       if(zi(ie,je).eq.-1.0) then
          go to 330
       else
c
c      ***********************************************************************
c
c      calculate the lag distance between zi(i,j) and each zi(ie,je) along ray
c      and use this value as a counter for the correlation analysis.  Round up
c      or round down
c
c      ***********************************************************************
c
          zz=sqrt(((ie-i)**2.)+((je-j)**2.))
          jj=sqrt(((ie-i)**2.)+((je-j)**2.))
          zzz=zz-jj
       if(zzz.ge.0.5) then
          jj=jj+1
       end if
c
       if(jj.le.105) then
```

```
              t1(jj)=((zi(i,j)-avedepth)*(zi(ie,je)-avedepth))+t1(jj)
              b1(jj)= ((zi(i,j)-avedepth)**2.)+b1(jj)
              b2(jj)=((zi(ie,je)-avedepth)**2.)+b2(jj)
              pairs(jj)=pairs(jj)+1
          end if
c
          end if
330       continue
310       continue
c
c
180       continue
          print,'row ',j,' is done'
170       continue
c
c         ***********************************************************************
c
c         determine the spatial correlation corfficient at this lag distance
c
c         ***********************************************************************
c
345       continue
c
          do 340 jj=0,105
          if(jj.eq.0) then
              y(1)=1.0
              x(1)=0.
          else
              y(jj+1)=t1(jj)/(sqrt(b1(jj)*b2(jj)))
              x3=jj
              x4=x3/5
              x(jj+1)=x4
          end if
c
c         ***********************************************************************
c
c          draw the zero line for the plot legend
c
c         ***********************************************************************
c
              x2(jj+1)=x4
              y2(jj+1)=0.
          write(n1n,346)x(jj+1),y(jj+1),pairs(jj)
340       continue
346       format(2x,f4.1,f8.4,f13.1)
c
              k10=105
c
c
          if(choice.ne.1) then
              go to 35
          end if
c
c         this section of the program starts the disspla plotting routines
c         for the correlation function
c
          print,'entering the plotting routine for the correlation function'
          call comprs
          call blowup(0.9)
          call page(12.2222,9.4444)
```

```
      call basalf('stand')
      call mx4alf('greek',1h*)
      call yaxang(0.0)
      call xname('Spatial Lag *n)  (kilometers)$',100)
      call yname('Correlation Coefficient: *r(n))$',100)
      call area2d(9.0,5.0)
      call headin('Spatial Correlation$',100,1.5,1)
      call messag('Storm Day$',9,7.1,5.55)
      call messag(amnth,4,6.9,5.25)
      call messag(',$',1,7.8,5.25)
      call intno(iiday,7.5,5.25)
      call intno(iiyear,7.9,5.25)
      call thkfrm(0.01)
      call frame
      call graf(0.,3.,21,-1.,0.2,1.0)
      call poly3
      call curve(x,y,k10,0)
      call thkcrv(1)
      call curve(x2,y2,k10,0)
      call endgr(0)
      call endpl(0)
      call donepl
c
900   format('storm day ',i2,'/',i2,'/',i2)
c
35     continue
      end
```

```
c                       STORMDAY.FORTRAN
c
c     The program executes all the algorithmns necessary to analyze the
c     coarse mesh random field of total rainfall depth at Walnut Gulch, Az.
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ...............................................................
c
c                     definition of variables
c
c                 input variables
c
c     iwat      watershed i.d. number
c     igage     raingage number
c     imonth    month
c     iday      day
c     iyr       year
c     zd        total station storm day rainfall  (mm.)
c     xd        station spatial coordinate
c     yd        station spatial coordinate
c
c                  program variables
c
c     amonth    storm day month
c     amnth     storm day month
c     avegage   average station depth: E(Yo)  (mm.)
c     vargage   variance of station depth: VAR(Yo)  (sq. mm.)
c     stdevgag  standard deviation of station depth  (mm.)
c     zi        node total depth  (mm.)
c     xi        node spatial coordinate  (0.1 km.)
c     yi        node spatial coordinate  (0.1 km.)
c     nxi       number of columns in the coarse mesh matrix
c     nyi       number of rows in the coarse mesh matrix
c     nxi2      number of columns in the fine mesh matrix
c     nyi2      number of rows in the fine mesh matrix
c
c                       Moments
c
c     avedepth  average depth of the random field: E(Y)  (mm.)
c     vardepth  variance of the depth of the random field: VAR(Y)  (sq. mm.)
c     skwdpth   coefficient of skewness of the depth of the random field
c                    C.S.(Y)  (dimensionless)
c
c                   Spatial correlation
c
c     lag1      lag distance  (km.)
c     corr1     spatial correlation of the random field at a particular lag
c
c                   Variance function
c
c     area      area of the element in the variance function  (sq. km.)
c                    (variable dimension)
c     numpts    number of nodes comprising the variance function element
c     totdpth   summation of node point depths in each variance function
c                    element  (mm.)
c     eledpth   depth of the variance function finite element  (mm.)
c     aveeldpth average depth of the varinace funtion elements of
```

```
c                     particular area:E(Y k)  (mm.)
c       vareldpth variance of the variance funtion elements of
c                     particular area:VAR(Y k)  (sq. mm.)
c       gamma     variance function evaluated at element size of a
c                     particular area   (dimensionless)
c       xarea     plotting variable of size area  (sq. km.)
c       ygamma    plotting variable of the variance function
c
c                    Basin Boundary
c
c       zj        matrix which creates a two level horizontal plane for
c                     purpose of drawing a basin boundary
c
c       ............................................................
c
c
        dimension   iwat(93),igage(93),iyear(93),imonth(93)
        dimension   iday(93),zd(93),xd(93),yd(93),a(31)
        dimension   xi(147),yi(70),wk(558),zi(147,70)
        dimension   iwk(13173),amonth(6),zj(147,70)
        dimension   xarea(31),ygamma(31),area(0:30),gamma(0:30)
        dimension   eledpth(5000),aveeldph(0:30),vareldph(0:30)
        dimension   corr1(31),lag1(31),xzero(31),yzero(31),alpha(100)
        dimension   l1(1),xkey1(26),ykey1(26),xkey2(26),ykey2(26)
        dimension   xkey3(26),ykey3(26)
c
        common work(40000)
        real lag1,lag,l1
c
        data  (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
        data pi/3.14159265389793/
c
            api=pi
c
        print,'Are you linked to the IMSL and DISSPLA  libraries?'
c
        print,'enter the number of input records, currently 93'
        input,nd
c
c       **************************************************************************
c
c       the section reads in the storm data from file84 and divides
c       these coordinates in two (both x and y) for use in the coarse
c       mesh random field.
c
c       **************************************************************************
c
        do 30 i=1,nd
        read(84,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
       &xd(i),yd(i)
             xd(i)=xd(i)/2.
             yd(i)=yd(i)/2.
 30     continue
 c
 1000   format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
 c
             iiday=iday(1)
             iiyear=1900+iyear(1)
             iii=imonth(1)-5
        print,'..........................  '
```

```
      print1005,amonth(iii),iiday,iiyear
      print,'............................  '
1005  format('storm day ',a4,i3,i5/)
c
          amnth=amonth(iii)
c
c     ************************************************************************
c
c     calculate the average gauge depth: E(Yo)
c
c     ************************************************************************
c
      do 40 i=1,nd
          a5=zd(i)+a5
          d5=d5+1
40    continue
c
          avegage=a5/d5
c
c
c     ************************************************************************
c
c     this portion will calculate the variance of the gauge depth:
c     VAR(Yo)
c
c     ************************************************************************
c
          l=0
c
      do 50 i=1,nd
          ll=ll+1
          a4=((zd(i)-avegage)**2.)+a4
50    continue
          vargage=a4/(ll-1)
          stdevgag=sqrt(vargage)
      print 1010,avegage
      print 1020,stdevgag
1010  format('the average gauge depth=',f10.3,' mm.'/)
1020  format('the std. deviat. of the gauge depth=',f10.3,' mm squared'/)
c
c     reset variables equal to zero
c
          a5=0.0
          d5=0.0
          a4=0.0
c
c     ************************************************************************
c
c     this portion generates the x and y coordinates of the nodes in
c     coarse mesh matrix: xi and yi
c
c     ************************************************************************
c
      nxi=147
      nyi=70
      izi=147
      nxi1=146
      nyi1=69
      nxi2=nxi*2
      nyi2=nyi*2
      do 60 j=1,nxi
         xi(j)=j
60    continue
      do 70 i=1,nyi
         yi(i)=i
70    continue
c
c
c     ***********************************************************************
c
c     invoke the IMSL bivariate surface interpolator subroutine IQHSCV
c     to generate the coarse total rainfall depth surface
c
c     ***********************************************************************
c
c
      call iqhscv(xd,yd,zd,nd,xi,nxi,yi,nyi,zi,izi,iwk,wk,ier)
c
c
c     ***********************************************************************
c
c     this section of the program creates the numerical watershed boundary
c     filter.  It assigns  the nodes of the coarse mesh matrix outside of
c     the Walnut Gulch basin random field with depth values of -1. mm.
c     The values generated by bivariate surface interpolator for nodes
c     within the random field are left intact and untouched.
c
c     ***********************************************************************
c
c
c     nxi2  refers to the number of columns in the matrix
c     nyi2  refers to the number of rows in the matrix
c
      do 10 i=2,nyi2,2
      do 20 j=2,nxi2,2
c
         l=j/2
         m=i/2
c
c      These logic statements define the coarse mesh basin boundary.
c
       if((i.eq.2)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.4)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.6)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.8)) then
          zi(l,m)=-1.
          go to 20
       else if((i.eq.10).and.(((j.ge.126).and.(j.le.128)).or.((j.ge.148).and.
     &(j.le.150)))) then
          go to 25
       else if(i.eq.10) then
          zi(l,m)=-1.
       else if((i.eq.12).and.(((j.ge.88).and.(j.le.90)).or.((j.ge.126)
     &.and.(j.le.132)).or.((j.ge.146).and.(j.le.150)))) then
```

```
         go to 25
      else if(i.eq.12) then
         zi(1,m)=-1.
      else if((i.eq.14).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126)
     &.and.(j.le.138)).or.((j.ge.140).and.(j.le.152)))) then
         go to 25
      else if(i.eq.14) then
         zi(1,m)=-1.
      else if((i.eq.16).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.116).and.
     &(j.le.160)))) then
         go to 25
      else if(i.eq.16) then
         zi(1,m)=-1.
      else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
     &(j.le.162)))) then
         go to 25
      else if(i.eq.18) then
         zi(1,m)=-1.
      else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.116).and.
     &(j.le.164)))) then
         go to 25
      else if(i.eq.20) then
         zi(1,m)=-1.
      else if((i.eq.22).and.(((j.ge.80).and.(j.le.82)).or.((j.ge.84).and.
     &(j.le.102)).or.((j.ge.118).and.(j.le.166)))) then
         go to 25
      else if(i.eq.22) then
         zi(1,m)=-1.
      else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
     &(j.le.166)))) then
         go to 25
      else if(i.eq.24) then
         zi(1,m)=-1.
      else if((i.eq.26).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.118).and.
     &(j.le.166)))) then
         go to 25
      else if(i.eq.26) then
         zi(1,m)=-1.
      else if((i.eq.28).and.(((j.ge.84).and.(j.le.106)).or.((j.ge.118).and.
     &(j.le.170)))) then
         go to 25
      else if(i.eq.28) then
         zi(1,m)=-1.
      else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.110).and.
     &(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
         go to 25
      else if(i.eq.30) then
         zi(1,m)=-1.
         go to 25
      else if((i.eq.32).and.(j.ge.82).and.(j.le.170)) then
         go to 25
      else if(i.eq.32) then
         zi(1,m)=-1.
      else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
         go to 25
      else if(i.eq.34) then
         zi(1,m)=-1.
         go to 25
      else if((i.eq.36).and.(j.ge.82).and.(j.le.174)) then
         go to 25
```

```
       else if(i.eq.36) then
          zi(1,m)=-1.
       else if((i.eq.38).and.(j.ge.84).and.(j.le.174)) then
          go to 25
       else if(i.eq.38) then
          zi(1,m)=-1.
       else if((i.eq.40).and.(j.ge.82).and.(j.le.172)) then
          go to 25
       else if(i.eq.40) then
          zi(1,m)=-1.
       else if((i.eq.42).and.(((j.ge.52).and.(j.le.54)).or.((j.ge.70).and.
     &(j.le.74)).or.((j.ge.78).and.(j.le.172)))) then
          go to 25
       else if(i.eq.42) then
          zi(1,m)=-1.
       else if((i.eq.44).and.(((j.ge.52).and.(j.le.56)).or.((j.ge.58).and.
     &(j.le.172)))) then
          go to 25
       else if(i.eq.44) then
          zi(1,m)=-1.
       else if((i.eq.46).and.(j.ge.52).and.(j.le.174)) then
          go to 25
       else if(i.eq.46) then
          zi(1,m)=-1.
       else if((i.eq.48).and.(j.ge.54).and.(j.le.174)) then
          go to 25
       else if(i.eq.48) then
          zi(1,m)=-1.
       else if((i.eq.50).and.(((j.ge.58).and.(j.le.174)).or.((j.ge.180).and.
     &(j.le.182)))) then
          go to 25
       else if(i.eq.50) then
          zi(1,m)=-1.
       else if((i.eq.52).and.(j.ge.58).and.(j.le.182)) then
          go to 25
       else if(i.eq.52) then
          zi(1,m)=-1.
       else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
          go to 25
       else if(i.eq.54) then
          zi(1,m)=-1.
       else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
          go to 25
       else if(i.eq.56) then
          zi(1,m)=-1.
       else if((i.eq.58).and.(((j.ge.36).and.(j.le.38)).or.((j.ge.52).and.
     &(j.le.196)))) then
          go to 25
       else if(i.eq.58) then
          zi(1,m)=-1.
       else if((i.eq.60).and.(j.ge.34).and.(j.le.198)) then
          go to 25
       else if(i.eq.60) then
          zi(1,m)=-1.
       else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
          go to 25
       else if(i.eq.62) then
          zi(1,m)=-1.
       else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
     &(j.le.221)))) then
```

```
    go to 25
else if(i.eq.64) then
    zi(1,m)=-1.
else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
    go to 25
else if(i.eq.66) then
    zi(1,m)=-1.
else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
    go to 25
else if(i.eq.68) then
    zi(1,m)=-1.
else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
    go to 25
else if(i.eq.70) then
    zi(1,m)=-1.
else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
    go to 25
else if(i.eq.72) then
    zi(1,m)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 25
else if(i.eq.74) then
    zi(1,m)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 25
else if(i.eq.76) then
    zi(1,m)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 25
else if(i.eq.78) then
    zi(1,m)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 25
else if(i.eq.80) then
    zi(1,m)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 25
else if(i.eq.82) then
    zi(1,m)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 25
else if(i.eq.84) then
    zi(1.m)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 25
else if(i.eq.86) then
    zi(1,m)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.88) then
    zi(1,m)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.238)) then
    go to 25
else if(i.eq.90) then
    zi(1,m)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 25
else if(i.eq.92) then
    zi(1,m)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
```

```
          go to 25
       else if(i.eq.94) then
          zi(l,m)=-1.
       else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
          go to 25
       else if(i.eq.96) then
          zi(l,m)=-1.
       else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
          go to 25
       else if(i.eq.98) then
          zi(l,m)=-1.
       else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
          go to 25
       else if(i.eq.100) then
          zi(l,m)=-1.
       else if((i.eq.102).and.(j.ge.44).and.(j.le.239)) then
          go to 25
       else if(i.eq.102) then
          zi(l,m)=-1.
       else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
          go to 25
       else if(i.eq.104) then
          zi(l,m)=-1.
       else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
          go to 25
       else if(i.eq.106) then
          zi(l,m)=-1.
       else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
          go to 25
       else if(i.eq.108) then
          zi(l,m)=-1.
       else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
          go to 25
       else if(i.eq.110) then
          zi(l,m)=-1.
       else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
          go to 25
       else if(i.eq.112) then
          zi(l,m)=-1.
       else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
     &.and.(j.le.256)))) then
          go to 25
       else if(i.eq.114) then
          zi(l,m)=-1.
       else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
     &.and.(j.le.257)))) then
          go to 25
       else if(i.eq.116) then
          zi(l,m)=-1.
       else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
     &.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
          go to 25
       else if(i.eq.118) then
          zi(l,m)=-1.
       else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
     &.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
          go to 25
       else if(i.eq.120) then
          zi(l,m)=-1.
       else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.273)
     &.and.(j.le.283)))) then
         go to 25
      else if(i.eq.122) then
          zi(l,m)=-1.
      else if((i.eq.124).and.(((j.ge.243).and.(j.le.258)).or.((j.ge.272)
     &.and.(j.le.281)))) then
         go to 25
      else if(i.eq.124) then
          zi(l,m)=-1.
      else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
     &.and.(j.le.279)))) then
         go to 25
      else if(i.eq.126) then
          zi(l,m)=-1.
      else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
         go to 25
      else if(i.eq.128) then
          zi(l,m)=-1.
      else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
         go to 25
      else if(i.eq.130) then
          zi(l,m)=-1.
      else if(i.eq.132) then
          zi(l,m)=-1.
         go to 20
      else if(i.eq.134) then
         zi(l,m)=-1.
         go to 20
      else if(i.eq.136) then
         zi(l,m)=-1.
         go to 20
      else if(i.eq.138) then
         zi(l,m)=-1.
         go to 20
      else if(i.eq.140) then
         zi(l,m)=-1.
         go to 20
      end if
c
25    continue
20    continue
10    continue
c
c     ************************************************************************
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, and imposes the
c     minimum depth filter where any node in the random field with a
c     value of less than or equal to 0.01 mm. is reassigned a value of
c     0.0 mm
c
c     ************************************************************************
c
      do 80 i=1,nxi
      do 90 j=1,nyi
      if(zi(i,j).eq.-1.) then
          go to 90
      else if(zi(i,j).le.0.01) then
          zi(i,j)=0.0
      end if
```

```
90    continue
80    continue
c
c
c     ***************************************************************************
c
c     calculate the average rainfall depth:E(Y)  in the random field
c
c     ***************************************************************************
c
c
      do 110 i=1,nxi
      do 120 j=1,nyi
c
      if(zi(i,j).eq.-1.)  then
          go to 120
      end if
          a1=zi(i,j)+a1
          d1=d1+1
 120  continue
 110  continue
          avedepth=a1/d1
c
c     ***************************************************************************
c
c     this portion will calculate the variance of the rainfall depth: VAR(Y)
c     and the coefficient of skewness of the rainfall depth: C.S.(Y) in the
c     random field.
c
c     ***************************************************************************
c
c
          l=0
c
      do 130 i=1,nxi
      do 140 j=1,nyi
c
      if(zi(i,j).eq.-1.)  then
          go to 140
      end if
          l=l+1
          a2=((zi(i,j)-avedepth)**2.)+a2
          a3=((zi(i,j)-avedepth)**3.)+a3
 140  continue
 130  continue
          vardepth=a2/(l-1)
          stdevdph=sqrt(vardepth)
          skwdpth=a3/((l-1)*(stdevdph**3.0))
      print 1030,avedepth
      print 1040,vardepth
      print 1045,skwdpth
 1030 format('the average point depth=',f10.4,' mm.')
 1040 format('the variance of the point depth=',f10.4,' mm ')
 1045 format('the coefficeint of skewness of the point depth=',f10.4)
c
c     reset variables equal to zero
c
          a1=0.0
          a2=0.0
          a3=0.0
```

```
c
c     ***************************************************************************
c
c     this portion of the program will determine the two dimensional
c     spatial correlation of the random field.  The technique used is
c     a bi-directional sampling technique.   The correlation will be
c     determined for lag distances which range from 0.0 to 6.0 km.
c
c     ***************************************************************************
c
c     entering the incremental (changes spatial distance) loop
c
      do 150 k1=0,30
c
          t1=0.
          b1=0.
          b2=0.
          d2=0.
          nxi3=nxi-k1
          nyi3=nyi-k1
c
c     this part of the correlation determination sweeps from west to
c     east across the random field.
c
      do 160 j=1,nyi
      do 170 i=1,nxi3
      if((zi(i,j).eq.-1.).or.(zi(i+k1,j).eq.-1.)) then
          go to 170
      else
          t1=((zi(i,j)-avedepth)*(zi(i+k1,j)-avedepth))+t1
          b1=((zi(i,j)-avedepth)**2.)+b1
          b2=((zi(i+k1,j)-avedepth)**2.)+b2
          d2=d2+1
      end if
170   continue
160   continue
c
c     this part of the correlation determination sweeps from the south
c     to the north across the random field
c
      do 180 i=1,nxi
      do 190 j=1,nyi3
      if((zi(i,j).eq.-1.).or.(zi(i,j+k1).eq.-1.)) then
          go to 190
      else
          t1=((zi(i,j)-avedepth)*(zi(i,j+k1)-avedepth))+t1
          b1=((zi(i,j)-avedepth)**2.)+b1
          b2=((zi(i,j+k1)-avedepth)**2.)+b2
          d2=d2+1
      end if
190   continue
180   continue
c
c     calculate the correlation coefficient for this spacing
c
          e1=0
          e1=t1/(sqrt(b1)*sqrt(b2))
          corr1(k1+1)=e1
          r1=k1
          r2=r1/5
          lag1(k1+1)=r2
c        draw the zero line
          xzero(k1+1)=r2
          yzero(k1+1)=0.
c
150     continue
c
c       reset variables equal to zero
c
          t1=0.0
          b1=0.0
          b2=0.0
          d2=0.0
          e1=0.0
          r1=0.0
          r2=0.0
          k1=0
c
c       ***************************************************************************
c
c       this portion of the program samples the coearse mesh random field for
c       function: gamma(area) beginning with elements of are 0.0 sq. km. to
c       elements of 36 sq. km.  The elements overlap are overlapped in the
c       technique.
c
c       ***************************************************************************
c
c       begin the loop which varies the length scale of the element
c
        do 220 m=0,30
c
           area(m)=((m*0.2)**2.)
c
c       enter the random field at this area.  Sweep the mesh from west to east
c       and then south to north simultaneously.
c
        do 230 j=1,nyi-m
        do 240 i=1,nxi-m
c
c       trip the element dimension counter
c
           k1=k1+1
c
c       determine the depth of the individual element by averaging the total
c       rainfall depth of each of the nodes comprising it.
c
        do 250 l=j,j+m
        do 260 k=i,i+m
c
c       check to see if any of the node depths lie outside the random field
c       i.e. equal -1.0 mm.
c
        if(zi(k,l).eq.-1.) then
           numpts=0
           totdpth=0.
           k1=k1-1
           go to 240
        else
           numpts=numpts+1
           totdpth=totdpth+zi(k,l)
```

```
      end if
c
260   continue
250   continue
c
c      the element depth is the average value of the point depths
c      which comprise the element
c
          eledpth(k1)=totdpth/numpts
          toteldph=toteldph+eledpth(k1)
          numpts=0
          totdpth=0.0
c
240    continue
230    continue
c
c      determine the average element depth: E(Y k) for this area
c
          aveeldph(m)=toteldph/k1
c
c      determine the variance of the element depth: VAR(Y k) for this area
c
       do 270 k2=1,k1
c
          totvardp=totvardp+((eledpth(k2)-aveeldph(m))**2.)
c
270    continue
c
          vareldph(m)=totvardp/(k1-1)
c
c      determine the value of gamma(A).  Reset all values of gamma(A)
c      greater than one equal to one for plotting purposes only.
c
          gamma(m)=vareldph(m)/vardepth
          xarea(m+1)=area(m)
       if(gamma(m).gt.1.0) then
          ygamma(m+1)=1.0
       else
          ygamma(m+1)=gamma(m)
       end if
          toteldph=0.
          totvardp=0.
          k1=0
       print221,area(m),gamma(m)
 220   continue
 221   format('area=',f5.2,' gamma(A)=',f7.4)
c
c      reset variables equal to zero
c
          toteldph=0.0
          totvardp=0.0
          b1=0.0
          b2=0.0
          b3=0.0
c
c      ***************************************************************************
c
c      This portion of the program will blank out the area outside of the
c      watershed basin boundary for the purpose of creating a basin boundary
c      for plotting purposes
```

```
c
c     ************************************************************************
c
      do 390 i=1,nxi
      do 410 j=1,nyi
c
      if(zi(i,j).eq.-1.0) then
         zj(i,j)=0.1
      else
         zj(i,j)=-0.99
      end if
c

410   continue
390   continue
c
c     ************************************************************************
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, this second filter
c     is only for plotting purposes
c
c     ************************************************************************
c
c
      do 420 i=1,nxi
      do 430 j=1,nyi
c
      if((zi(i,j).lt.0.01).and.(zi(i,j).gt.-1.0)) then
         zi(i,j)=-0.1
      else if((zi(i,j).eq.-1.)) then
         zi(i,j)=0.0
      end if
c
430   continue
420   continue
c
c     ************************************************************************
c
c     Write the storm day date, the first three moments of the random field
c     and the numerical spatial correlation and variance function results
c     to file 91
c
c     ************************************************************************
c
      write (91,1160)iyear(1),imonth(1),iday(1)
      write (91,1170)avedepth,vardepth,skwdpth
      do 460 i=1,31
      write (91,1180)lag1(i),corr1(i),area(i-1),gamma(i-1)
460   continue
1160  format(3i2)
1170  format(3(1x,f7.3))
1180  format(f5.2,1x,f7.3,1x,f6.2,1x,f6.3)
c
c
c     ************************************************************************
c
c     this portion of the code draws straight line segments to be
c     plotted on the legend (coutour lines and basin boundary)
c
```

```
c     ************************************************************************
c
      do 490 j=208,233
c
         i=j-207
         r10=j
         r11=r10/10.0
         xkey1(i)=r11
         ykey1(i)=3.4
         xkey2(i)=r11
         ykey2(i)=2.8
         xkey3(i)=r11
         ykey3(i)=1.8
c
490   continue
c
c
c     ************************************************************************
c
c     this section of the program starts the disspla plotting routines
c     for the correlation function
c
c     ************************************************************************
c
c
      print,'entering the plotting routine for the correlation function'
      call comprs
      call blowup(0.6063)
      call page(14.0196,18.1428)
      call physor(2.98688,3.1338)
      call basalf('stand')
      call mx4alf('greek',1h*)
      call yaxang(0.0)
      call xname('Spatial Lag *n) (kilometers)$',100)
      call yname('Correlation Coefficient: *r(n))$',100)
      call area2d(4.286,4.286)
      call headin('Spatial Correlation$',100,1.5,1)
      call thkfrm(0.01)
      call frame
      call graf(0.,1.,6.,-1.,0.2,1.0)
      call poly3
      call curve(lag1,corr1,31,0)
      call thkcrv(i)
      call curve(xzero,yzero,31,0)
      call endgr(1)
c
c     ************************************************************************
c
c     this section of the program starts the disspla plotting routines
c     for the variance function
c
c     ************************************************************************
c
      print,'entering the variance function plotting routine'
      call physor(8.7009,3.1338)
      call xname('Area (square kilometers)$',100)
      call yname('Variance Function: *g) (Area)$',100)
      call area2d(4.286,4.286)
      call headin('Variance Function$',100,1.5,1)
      call thkfrm(0.01)
      call frame
      call graf(0.,4.,36.,0.,0.1,1.0)
      call poly3
      call curve(xarea,ygamma,31,0)
      call endgr(2)
c
c     ***********************************************************************
c
c     entering the disspla routine for the contour plot
c
c     ***********************************************************************
c
      print,'enter the countour plot plotting routine'
c
      call physor(2.9869,9.8136)
      call area2d(10.0,4.762)
      call reset('yaxang')
      call xname('Kilometers$',100)
      call yname('Kilometers$',100)
      call headin('Walnut Gulch, Arizona$',100,1.5,2)
      call headin('Ac=154.21 sq.km.$',100,1.0,2)
      call intaxs
      call xintax
      call yintax
      call messag('Contour Interval: 2 mm.$',100,0.3,4.25)
      call messag('Storm Day$',9,8.1,5.5)
      call messag(amonth(iii),4,7.9,5.2)
      call messag(',$',1,8.8,5.2)
      call intno(iiday,8.5,5.2)
      call intno(iiyear,8.9,5.2)
      call messag('E(Yo)=',6,0.45,0.8)
      call realno(avegage,2,1.3,0.8)
      call messag('mm.',3,2.0,0.8)
      call messag('S.D.(Yo)=',9,0.1,0.5)
      call realno(stdevgag,2,1.3,0.5)
      call messag('mm.',3,2.0,0.5)
      call messag('Legend',6,7.9,1.5)
      call messag('Isohyet Line',12,8.0,1.1)
      call messag('Catchment',9,8.0,0.7)
      call messag('Boundary',8,8.0,0.5)
      call graf(0.,2.0,30.0,0.,1.0,14.0)
      call thkfrm(0.01)
      call frame
      call curve(xkey1,ykey1,26,0)
      call dash
      call curve(xkey2,ykey2,26,0)
      call reset('dash')
      call dot
      call thkcrv(5)
      call curve(xkey3,ykey3,26,0)
      call reset('dot')
      call bcomon(40000)
      call conthn(0.04)
c
c     plot the contours
c
      call conmak(zi,nxi,nyi,2.)
      call conlin(0,'solid','label',1,7)
      call conlin(1,'dash','labels',1,4)
      call conang(90.)
      call raspln(0.25)
      print,'entering the total depth isohyet drawing routine'
      call contur(2,'labels','draw')
c
c     scribe the outline of the basin boundary as a separate contour line
c
      call reset('conang')
      call reset('conthn')
      call bcomon(40000)
      call conmak(zj,nxi,nyi,2.)
      call conlin(0,'dot','no labels',5,10)
      call raspln(0.25)
      print,'entering the basin boundary drawing routine'
      print,' *** this routine will require restarting, simply type'
      print,' "start" (without the quotation marks) and hit return ***'
      call contur(1,'nolabels','draw')
      call endpl(0)
      call donepl
c
      end
```

```
c                  STORMWET.FORTRAN
c
c     The program executes all the algorithmns necessary to analyze the fine
c     mesh random field of total rainfall depth at Walnut Gulch, Az.
c
c     This program was written by Neil M. Fennessey at M.I.T.
c     during the course of his Master's Degree research into the
c     Areal Distribution of Total Rainfall Depth.
c
c     ..............................................................
c
c                   definition of variables
c
c                input variables
c
c     iwat       watershed i.d. number
c     igage      raingage number
c     imonth     month
c     iday       day
c     iyr        year
c     zd         total station storm day rainfall (mm.)
c     xd         station spatial coordinate
c     yd         station spatial coordinate
c
c                 fine mesh random field variables
c
c     zi         node total depth (mm.)
c     xi         node spatial coordinate (0.1 km.)
c     yi         node spatial coordinate (0.1 km.)
c     nxi        number of columns in the fine mesh matrix
c     nyi        number of rows in the fine mesh matrix
c
c                 spatial distribution variables
c
c     d          total rainfall depth of the 0.01 sq. km. finite element
c     y          fraction of total area wetted to a depth Y greater than
c                      or equal to y
c     x          depth (mm.)
c     count      summation of elements at a particular depth x
c     totcount   summation of elements at a particular depth x
c     totcnt2    summation of elements at a particular depth x
c
c
c     ..................................................................
c
      dimension   count(-2:200),a(5),c(5)
      dimension   iwat(93),igage(93),iyear(93),imonth(93)
      dimension   iday(93),zd(93),xd(93),yd(93)
      dimension   xi(294),yi(140),wk(558),x(200),y(200)
      dimension   totcnt2(-1:200),zi(294,140)
c
      common /work1/iwk(44043)
      common /element1/d(25000)
c
      print,'Are you linked to the IMSL library?'
c
      print,'enter the number of input records, currently 93'
      input,nd
c
c     ***********************************************************************
```

```
c
c     the section reads in the storm day station observation data from file84
c
c     **************************************************************************
c
      do 30 i=1,nd
c
      read(84,1000)iwat(i),igage(i),iyear(i),imonth(i),iday(i),zd(i),
     &xd(i),yd(i)
30    continue
c
1000  format(i2,i3,1x,3i2,f6.2,2(1x,f4.0))
c
      print,'...........................  '
      print1010,imonth(1),iday(1),iyear(1)
      print,'...........................  '
c
c     **************************************************************************
c
c     this portion generates the x and y coordinates of the nodes in
c     fine mesh matrix: xi and yi
c
c     **************************************************************************
c
      nxi=294
      nyi=140
      izi=nxi
      nxi1=293
      nyi1=139
c
      do 40 j=1,nxi
          xi(j)=j
40    continue
c
      do 50 k=1,nyi
          yi(k)=k
50    continue
c
c     **************************************************************************
c
c     invoke the IMSL bivariate surface interpolator subroutine IQHSCV
c     to generate the coarse total rainfall depth surface
c
c     **************************************************************************
c
      call iqhscv(xd,yd,zd,nd,xi,nxi,yi,nyi,zi,izi,iwk,wk,ier)
c
c     **************************************************************************
c
c     this section of the program creates the numerical watershed boundary
c     filter.  It assigns  the nodes of the fine mesh matrix outside of
c     the Walnut Gulch basin random field with depth values of -1.0 mm.
c     The values generated by bivariate surface interpolator for nodes
c     within the random field are left intact and untouched.
c
c     **************************************************************************
c
c     nxi refers to the number of columns in the matrix
c     nyi refers to the number of rows in the matrix
c
```

```
      do 10 i=1,nyi
      do 20 j=1,nxi
c
      if(j.eq.nxi+1) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.1)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.2)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.3)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.4)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.5)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.6)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.7)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.8)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.9)) then
          zi(j,i)=-1.
          go to 20
      else if((i.eq.10).and.(((j.ge.127).and.(j.le.129)).or.((j.ge.149).and.
     &(j.le.150)))) then
          go to 25
      else if(i.eq.10) then
          zi(j,i)=-1.
      else if((i.eq.11).and.(((j.ge.127).and.(j.le.132)).or.((j.ge.148).and.
     &(j.le.151)))) then
          go to 25
      else if(i.eq.11) then
          zi(j,i)=-1.
      else if((i.eq.12).and.(((j.ge.89).and.(j.le.90)).or.((j.ge.127)
     &.and.(j.le.133)).or.((j.ge.147).and.(j.le.151)))) then
          go to 25
      else if(i.eq.12) then
          zi(j,i)=-1.
      else if((i.eq.13).and.(((j.ge.89).and.(j.le.91)).or.((j.ge.128)
     &.and.(j.le.135)).or.((j.ge.140).and.(j.le.141)).or.
     &((j.ge.143).and.(j.le.152)))) then
          go to 25
      else if(i.eq.13) then
          zi(j,i)=-1.
      else if((i.eq.14).and.(((j.ge.89).and.(j.le.94)).or.((j.ge.127)
     &.and.(j.le.138)).or.((j.ge.139).and.(j.le.153)))) then
          go to 25
      else if(i.eq.14) then
          zi(j,i)=-1.
      else if((i.eq.15).and.(((j.ge.88).and.(j.le.94)).or.((j.ge.126).and.
&(j.le.158)))) then
      go to 25
 else if(i.eq.15) then
      zi(j,i)=-1.
 else if((i.eq.16).and.(((j.ge.88).and.(j.le.95)).or.((j.ge.116).and.
&(j.le.161)))) then
      go to 25
 else if(i.eq.16) then
      zi(j,i)=-1.
 else if((i.eq.17).and.(((j.ge.88).and.(j.le.99)).or.((j.ge.116).and.
&(j.le.161)))) then
      go to 25
 else if(i.eq.17) then
      zi(j,i)=-1.
 else if((i.eq.18).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.116).and.
&(j.le.162)))) then
      go to 25
 else if(i.eq.18) then
      zi(j,i)=-1.
 else if((i.eq.19).and.(((j.ge.88).and.(j.le.100)).or.((j.ge.117).and.
&(j.le.162)))) then
      go to 25
 else if(i.eq.19) then
      zi(j,i)=-1.
 else if((i.eq.20).and.(((j.ge.88).and.(j.le.102)).or.((j.ge.117).and.
&(j.le.164)))) then
      go to 25
 else if(i.eq.20) then
      zi(j,i)=-1.
 else if((i.eq.21).and.(((j.ge.81).and.(j.le.82)).or.((j.ge.86).and.
&(j.le.102)).or.((j.ge.119).and.(j.le.165)))) then
      go to 25
 else if(i.eq.21) then
      zi(j,i)=-1.
 else if((i.eq.22).and.(((j.ge.81).and.(j.le.83)).or.((j.ge.85).and.
&(j.le.103)).or.((j.ge.119).and.(j.le.166)))) then
      go to 25
 else if(i.eq.22) then
      zi(j,i)=-1.
 else if((i.eq.23).and.(((j.ge.81).and.(j.le.106)).or.((j.ge.119).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.23) then
      zi(j,i)=-1.
 else if((i.eq.24).and.(((j.ge.82).and.(j.le.106)).or.((j.ge.119).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.24) then
      zi(j,i)=-1.
 else if((i.eq.25).and.(((j.ge.83).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.25) then
      zi(j,i)=-1.
 else if((i.eq.26).and.(((j.ge.83).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.166)))) then
      go to 25
 else if(i.eq.26) then
      zi(j,i)=-1.
 else if((i.eq.27).and.(((j.ge.83).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.168)))) then
     go to 25
 else if(i.eq.27) then
     zi(j,i)=-1.
 else if((i.eq.28).and.(((j.ge.84).and.(j.le.107)).or.((j.ge.118).and.
&(j.le.171)))) then
     go to 25
 else if(i.eq.28) then
     zi(j,i)=-1.
 else if((i.eq.29).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.118).and.
&(j.le.172)))) then
     go to 25
 else if(i.eq.29) then
     zi(j,i)=-1.
 else if((i.eq.30).and.(((j.ge.84).and.(j.le.108)).or.((j.ge.111).and.
&(j.le.114)).or.((j.ge.116).and.(j.le.172)))) then
     go to 25
 else if(i.eq.30) then
     zi(j,i)=-1.
 else if((i.eq.31).and.(j.ge.83).and.(j.le.172)) then
     go to 25
 else if(i.eq.31) then
     zi(j,i)=-1.
 else if((i.eq.32).and.(j.ge.83).and.(j.le.171)) then
     go to 25
 else if(i.eq.32) then
     zi(j,i)=-1.
 else if((i.eq.33).and.(j.ge.82).and.(j.le.173)) then
     go to 25
 else if(i.eq.33) then
     zi(j,i)=-1.
 else if((i.eq.34).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.34) then
     zi(j,i)=-1.
 else if((i.eq.35).and.(j.ge.82).and.(j.le.174)) then
     go to 25
 else if(i.eq.35) then
     zi(j,i)=-1.
 else if((i.eq.36).and.(j.ge.83).and.(j.le.174)) then
     go to 25
 else if(i.eq.36) then
     zi(j,i)=-1.
 else if((i.eq.37).and.(j.ge.85).and.(j.le.174)) then
     go to 25
 else if(i.eq.37) then
     zi(j,i)=-1.
 else if((i.eq.38).and.(j.ge.85).and.(j.le.174)) then
     go to 25
 else if(i.eq.38) then
     zi(j,i)=-1.
 else if((i.eq.39).and.(j.ge.84).and.(j.le.174)) then
     go to 25
 else if(i.eq.39) then
     zi(j,i)=-1.
 else if((i.eq.40).and.(j.ge.82).and.(j.le.173)) then
     go to 25
 else if(i.eq.40) then
     zi(j,i)=-1.
 else if((i.eq.41).and.(j.ge.79).and.(j.le.173)) then
```

```
     go to 25
 else if(i.eq.41) then
     zi(j,i)=-1.
 else if((i.eq.42).and.(((j.ge.53).and.(j.le.54)).or.((j.ge.71).and.
&(j.le.75)).or.((j.ge.78).and.(j.le.172)))) then
     go to 25
 else if(i.eq.42) then
     zi(j,i)=-1.
 else if((i.eq.43).and.(((j.ge.53).and.(j.le.55)).or.((j.ge.60).and.
&(j.le.61)).or.((j.ge.71).and.(j.le.172)))) then
     go to 25
 else if(i.eq.43) then
     zi(j,i)=-1.
 else if((i.eq.44).and.(((j.ge.53).and.(j.le.57)).or.((j.ge.59).and.
&(j.le.172)))) then
     go to 25
 else if(i.eq.44) then
     zi(j,i)=-1.
 else if((i.eq.45).and.(j.ge.53).and.(j.le.173)) then
     go to 25
 else if(i.eq.45) then
     zi(j,i)=-1.
 else if((i.eq.46).and.(j.ge.53).and.(j.le.174)) then
     go to 25
 else if(i.eq.46) then
     zi(j,i)=-1.
 else if((i.eq.47).and.(j.ge.53).and.(j.le.174)) then
     go to 25
 else if(i.eq.47) then
     zi(j,i)=-1.
 else if((i.eq.48).and.(j.ge.55).and.(j.le.174)) then
     go to 25
 else if(i.eq.48) then
     zi(j,i)=-1.
 else if((i.eq.49).and.(j.ge.58).and.(j.le.174)) then
     go to 25
 else if(i.eq.49) then
     zi(j,i)=-1.
 else if((i.eq.50).and.(((j.ge.58).and.(j.le.175)).or.((j.ge.180).and.
&(j.le.182)))) then
     go to 25
 else if(i.eq.50) then
     zi(j,i)=-1.
 else if((i.eq.51).and.(((j.ge.58).and.(j.le.176)).or.((j.ge.180).and.
&(j.le.183)))) then
     go to 25
 else if(i.eq.51) then
     zi(j,i)=-1.
 else if((i.eq.52).and.(j.ge.58).and.(j.le.183)) then
     go to 25
 else if(i.eq.52) then
     zi(j,i)=-1.
 else if((i.eq.53).and.(j.ge.57).and.(j.le.183)) then
     go to 25
 else if(i.eq.53) then
     zi(j,i)=-1.
 else if((i.eq.54).and.(j.ge.57).and.(j.le.191)) then
     go to 25
 else if(i.eq.54) then
     zi(j,i)=-1.
 else if((i.eq.55).and.(j.ge.57).and.(j.le.192)) then
     go to 25
 else if(i.eq.55) then
     zi(j,i)=-1.
 else if((i.eq.56).and.(j.ge.54).and.(j.le.195)) then
     go to 25
 else if(i.eq.56) then
     zi(j,i)=-1.
 else if((i.eq.57).and.(j.ge.52).and.(j.le.196)) then
     go to 25
 else if(i.eq.57) then
     zi(j,i)=-1.
 else if((i.eq.58).and.(((j.ge.37).and.(j.le.39)).or.((j.ge.52).and.
&(j.le.196)))) then
     go to 25
 else if(i.eq.58) then
     zi(j,i)=-1.
 else if((i.eq.59).and.(((j.ge.35).and.(j.le.45)).or.((j.ge.47).and.
&(j.le.197)))) then
     go to 25
 else if(i.eq.59) then
     zi(j,i)=-1.
 else if((i.eq.60).and.(j.ge.35).and.(j.le.199)) then
     go to 25
 else if(i.eq.60) then
     zi(j,i)=-1.
 else if((i.eq.61).and.(j.ge.34).and.(j.le.203)) then
     go to 25
 else if(i.eq.61) then
     zi(j,i)=-1.
 else if((i.eq.62).and.(j.ge.34).and.(j.le.206)) then
     go to 25
 else if(i.eq.62) then
     zi(j,i)=-1.
 else if((i.eq.63).and.(((j.ge.34).and.(j.le.207)).or.((j.ge.219).and.
&(j.le.220)))) then
     go to 25
 else if(i.eq.63) then
     zi(j,i)=-1.
 else if((i.eq.64).and.(((j.ge.33).and.(j.le.208)).or.((j.ge.218).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.64) then
     zi(j,i)=-1.
 else if((i.eq.65).and.(((j.ge.33).and.(j.le.210)).or.((j.ge.217).and.
&(j.le.221)))) then
     go to 25
 else if(i.eq.65) then
     zi(j,i)=-1.
 else if((i.eq.66).and.(j.ge.32).and.(j.le.222)) then
     go to 25
 else if(i.eq.66) then
     zi(j,i)=-1.
 else if((i.eq.67).and.(j.ge.30).and.(j.le.223)) then
     go to 25
 else if(i.eq.67) then
     zi(j,i)=-1.
 else if((i.eq.68).and.(j.ge.30).and.(j.le.224)) then
     go to 25
 else if(i.eq.68) then
```

```
    zi(j,i)=-1.
else if((i.eq.69).and.(j.ge.29).and.(j.le.226)) then
    go to 25
else if(i.eq.69) then
    zi(j,i)=-1.
else if((i.eq.70).and.(j.ge.28).and.(j.le.226)) then
    go to 25
else if(i.eq.70) then
    zi(j,i)=-1.
else if((i.eq.71).and.(j.ge.27).and.(j.le.226)) then
    go to 25
else if(i.eq.71) then
    zi(j,i)=-1.
else if((i.eq.72).and.(j.ge.25).and.(j.le.226)) then
    go to 25
else if(i.eq.72) then
    zi(j,i)=-1.
else if((i.eq.73).and.(j.ge.23).and.(j.le.226)) then
    go to 25
else if(i.eq.73) then
    zi(j,i)=-1.
else if((i.eq.74).and.(j.ge.21).and.(j.le.226)) then
    go to 25
else if(i.eq.74) then
    zi(j,i)=-1.
else if((i.eq.75).and.(j.ge.19).and.(j.le.226)) then
    go to 25
else if(i.eq.75) then
    zi(j,i)=-1.
else if((i.eq.76).and.(j.ge.17).and.(j.le.226)) then
    go to 25
else if(i.eq.76) then
    zi(j,i)=-1.
else if((i.eq.77).and.(j.ge.15).and.(j.le.226)) then
    go to 25
else if(i.eq.77) then
    zi(j,i)=-1.
else if((i.eq.78).and.(j.ge.13).and.(j.le.226)) then
    go to 25
else if(i.eq.78) then
    zi(j,i)=-1.
else if((i.eq.79).and.(j.ge.12).and.(j.le.227)) then
    go to 25
else if(i.eq.79) then
    zi(j,i)=-1.
else if((i.eq.80).and.(j.ge.11).and.(j.le.228)) then
    go to 25
else if(i.eq.80) then
    zi(j,i)=-1.
else if((i.eq.81).and.(j.ge.10).and.(j.le.229)) then
    go to 25
else if(i.eq.81) then
    zi(j,i)=-1.
else if((i.eq.82).and.(j.ge.10).and.(j.le.230)) then
    go to 25
else if(i.eq.82) then
    zi(j,i)=-1.
else if((i.eq.83).and.(j.ge.11).and.(j.le.231)) then
    go to 25
else if(i.eq.83) then
```

```
    zi(j,i)=-1.
else if((i.eq.84).and.(j.ge.12).and.(j.le.233)) then
    go to 25
else if(i.eq.84) then
    zi(j,i)=-1.
else if((i.eq.85).and.(j.ge.13).and.(j.le.234)) then
    go to 25
else if(i.eq.85) then
    zi(j,i)=-1.
else if((i.eq.86).and.(j.ge.14).and.(j.le.234)) then
    go to 25
else if(i.eq.86) then
    zi(j,i)=-1.
else if((i.eq.87).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.87) then
    zi(j,i)=-1.
else if((i.eq.88).and.(j.ge.15).and.(j.le.235)) then
    go to 25
else if(i.eq.88) then
    zi(j,i)=-1.
else if((i.eq.89).and.(j.ge.16).and.(j.le.236)) then
    go to 25
else if(i.eq.89) then
    zi(j,i)=-1.
else if((i.eq.90).and.(j.ge.17).and.(j.le.236)) then
    go to 25
else if(i.eq.90) then
    zi(j,i)=-1.
else if((i.eq.91).and.(j.ge.17).and.(j.le.238)) then
    go to 25
else if(i.eq.91) then
    zi(j,i)=-1.
else if((i.eq.92).and.(j.ge.18).and.(j.le.238)) then
    go to 25
else if(i.eq.92) then
    zi(j,i)=-1.
else if((i.eq.93).and.(j.ge.19).and.(j.le.238)) then
    go to 25
else if(i.eq.93) then
    zi(j,i)=-1.
else if((i.eq.94).and.(j.ge.21).and.(j.le.238)) then
    go to 25
else if(i.eq.94) then
    zi(j,i)=-1.
else if((i.eq.95).and.(j.ge.23).and.(j.le.238)) then
    go to 25
else if(i.eq.95) then
    zi(j,i)=-1.
else if((i.eq.96).and.(j.ge.23).and.(j.le.238)) then
    go to 25
else if(i.eq.96) then
    zi(j,i)=-1.
else if((i.eq.97).and.(j.ge.24).and.(j.le.238)) then
    go to 25
else if(i.eq.97) then
    zi(j,i)=-1.
else if((i.eq.98).and.(j.ge.25).and.(j.le.239)) then
    go to 25
else if(i.eq.98) then
```

```
           zi(j,i)=-1.
       else if((i.eq.99).and.(j.ge.30).and.(j.le.240)) then
           go to 25
       else if(i.eq.99) then
           zi(j,i)=-1.
       else if((i.eq.100).and.(j.ge.35).and.(j.le.240)) then
           go to 25
       else if(i.eq.100) then
           zi(j,i)=-1.
       else if((i.eq.101).and.(j.ge.40).and.(j.le.240)) then
           go to 25
       else if(i.eq.101) then
           zi(j,i)=-1.
       else if((i.eq.102).and.(j.ge.45).and.(j.le.239)) then
           go to 25
       else if(i.eq.102) then
           zi(j,i)=-1.
       else if((i.eq.103).and.(j.ge.50).and.(j.le.241)) then
           go to 25
       else if(i.eq.103) then
           zi(j,i)=-1.
       else if((i.eq.104).and.(j.ge.67).and.(j.le.243)) then
           go to 25
       else if(i.eq.104) then
           zi(j,i)=-1.
       else if((i.eq.105).and.(j.ge.79).and.(j.le.245)) then
           go to 25
       else if(i.eq.105) then
           zi(j,i)=-1.
       else if((i.eq.106).and.(j.ge.88).and.(j.le.247)) then
           go to 25
       else if(i.eq.106) then
           zi(j,i)=-1.
       else if((i.eq.107).and.(j.ge.95).and.(j.le.251)) then
           go to 25
       else if(i.eq.107) then
           zi(j,i)=-1.
       else if((i.eq.108).and.(j.ge.104).and.(j.le.252)) then
           go to 25
       else if(i.eq.108) then
           zi(j,i)=-1.
       else if((i.eq.109).and.(((j.ge.106).and.(j.le.112)).or.((j.ge.115).and.
     &(j.le.117)).or.((j.ge.120).and.(j.le.253)))) then
           go to 25
       else if(i.eq.109) then
           zi(j,i)=-1.
       else if((i.eq.110).and.(j.ge.125).and.(j.le.254)) then
           go to 25
       else if(i.eq.110) then
           zi(j,i)=-1.
       else if((i.eq.111).and.(j.ge.127).and.(j.le.255)) then
           go to 25
       else if(i.eq.111) then
           zi(j,i)=-1.
       else if((i.eq.112).and.(j.ge.129).and.(j.le.255)) then
           go to 25
       else if(i.eq.112) then
           zi(j,i)=-1.
       else if((i.eq.113).and.(j.ge.131).and.(j.le.255)) then
           go to 25
```

```
      else if(i.eq.113) then
           zi(j,i)=-1.
      else if((i.eq.114).and.(((j.ge.135).and.(j.le.138)).or.((j.ge.140)
     &.and.(j.le.256)))) then
          go to 25
      else if(i.eq.114) then
           zi(j,i)=-1.
      else if((i.eq.115).and.(j.ge.147).and.(j.le.257)) then
          go to 25
      else if(i.eq.115) then
           zi(j,i)=-1.
      else if((i.eq.116).and.(((j.ge.150).and.(j.le.200)).or.((j.ge.210)
     &.and.(j.le.257)))) then
          go to 25
      else if(i.eq.116) then
           zi(j,i)=-1.
      else if((i.eq.117).and.(((j.ge.152).and.(j.le.196)).or.((j.ge.212)
     &.and.(j.le.257)))) then
          go to 25
      else if(i.eq.117) then
           zi(j,i)=-1.
      else if((i.eq.118).and.(((j.ge.153).and.(j.le.193)).or.((j.ge.214)
     &.and.(j.le.218)).or.((j.ge.225).and.(j.le.257)))) then
          go to 25
      else if(i.eq.118) then
           zi(j,i)=-1.
      else if((i.eq.119).and.(((j.ge.155).and.(j.le.191)).or.((j.ge.226)
     &.and.(j.le.230)).or.((j.ge.247).and.(j.le.258))
     &.or.((j.ge.277).and.(j.le.279))
     &.or.((j.ge.281).and.(j.le.283)))) then
          go to 25
      else if(i.eq.119) then
           zi(j,i)=-1.
      else if((i.eq.120).and.(((j.ge.158).and.(j.le.189)).or.((j.ge.237)
     &.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
          go to 25
      else if(i.eq.120) then
           zi(j,i)=-1.
      else if((i.eq.121).and.(((j.ge.170).and.(j.le.183)).or.((j.ge.238)
     &.and.(j.le.258)).or.((j.ge.277).and.(j.le.283)))) then
          go to 25
      else if(i.eq.121) then
           zi(j,i)=-1.
      else if((i.eq.122).and.(((j.ge.239).and.(j.le.258)).or.((j.ge.275)
     &.and.(j.le.283)))) then
          go to 25
      else if(i.eq.122) then
           zi(j,i)=-1.
      else if((i.eq.123).and.(((j.ge.240).and.(j.le.258)).or.((j.ge.273)
     &.and.(j.le.283)))) then
          go to 25
      else if(i.eq.123) then
           zi(j,i)=-1.
      else if((i.eq.124).and.(((j.ge.243).and.(j.le.259)).or.((j.ge.272)
     &.and.(j.le.281)))) then
          go to 25
      else if(i.eq.124) then
           zi(j,i)=-1.
      else if((i.eq.125).and.(((j.ge.245).and.(j.le.260)).or.((j.ge.269)
     &.and.(j.le.280)))) then
```

C-3

```
          go to 25
      else if(i.eq.125) then
          zi(j,i)=-1.
      else if((i.eq.126).and.(((j.ge.246).and.(j.le.264)).or.((j.ge.267)
     &.and.(j.le.279)))) then
          go to 25
      else if(i.eq.126) then
          zi(j,i)=-1.
      else if((i.eq.127).and.(j.ge.251).and.(j.le.276)) then
          go to 25
      else if(i.eq.127) then
          zi(j,i)=-1.
      else if((i.eq.128).and.(j.ge.253).and.(j.le.276)) then
          go to 25
      else if(i.eq.128) then
          zi(j,i)=-1.
      else if((i.eq.129).and.(j.ge.254).and.(j.le.276)) then
          go to 25
      else if(i.eq.129) then
          zi(j,i)=-1.
      else if((i.eq.130).and.(j.ge.268).and.(j.le.276)) then
          go to 25
      else if(i.eq.130) then
          zi(j,i)=-1.
      else if((i.eq.131).and.(j.ge.273).and.(j.le.275)) then
          go to 25
      else if(i.eq.131) then
          zi(j,i)=-1.
      else if(i.eq.132) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.133) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.134) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.135) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.136) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.137) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.138) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.139) then
          zi(j,i)=-1.
          go to 20
      else if(i.eq.140) then
          zi(j,i)=-1.
          go to 20
      end if
c
25    continue
20    continue
10    continue
```

```
c
c     ***************************************************************************
c
c     this portion of the program looks for undulations which may have
c     been created by the surface fitting subroutine, and imposes the
c     minimum depth filter where any node in the random field with a
c     value of less than or equal to 0.01 mm. is reassigned a value of
c     0.0 mm
c
c     ***************************************************************************
c
      do 60 i=1,nxi
      do 70 j=1,nyi
c
      if(zi(i,j).eq.-1.) then
          go to 70
      else if(zi(i,j).le.0.01) then
          zi(i,j)=0.0
      end if
c
70    continue
60    continue
c
c
c     ***************************************************************************
c
c     this portion of the program averages the 4 node depth values
c     at the corner of each finite area element in order to determine
c     the depth of the that element
c
c     ***************************************************************************
c
c     initialize counting variable k
c
          k=0
c
      do 130 j=1,nyil
          l5=0
      do 140 i=1,nxil
c
      if((zi(i,j).eq.-1.).or.(zi(i+1,j).eq.-1.).or.(zi(i,j+1).eq.-1.).or.
     &(zi(i+1,j+1).eq.-1.)) then
          go to 140
      end if
          k=k+1
          d(k)=(zi(i,j)+zi(i+1,j)+zi(i,j+1)+zi(i+1,j+1))/4.
c
140    continue
130    continue
c
          e=k
      print1040,e*.01
1040  format('total area in the entire basin=',f9.2,' sq.km.')
c
c     ***************************************************************************
c
c     this portion of the program will sift through all the element
c     depths and find the maximum value (truncated as an integer)
c
c     ***************************************************************************
```

```
c
      do 150 i=1,k-1
          a(2)=d(i)
          a(3)=d(i+1)
      if(a(2).gt.a(3)) then
          a(4)=a(2)
      else if (a(3).gt.a(2)) then
          a(4)=a(3)
      end if
      if(a(4).gt.a(1)) then
          a(1)=a(4)
      end if
          max=a(1)
          amax=max+1
150   continue
c
c     ************************************************************************
c
c      sum the elements at particular integer depths and group them
c      accordingly
c
c     ************************************************************************
c
      do 160 i=-2,max
      do 170 j=1,k
c
      if((d(j).lt.0.).and.(i.eq.-2)) then
          count(-2)=count(-2)+1
      else if((d(j).eq.0.).and.(i.eq.-1)) then
          count(-1)=count(-1)+1
      else if((d(j).gt.0.).and.(d(j).lt.1.).and.(i.eq.0).and.
     &(max.eq.0)) then
          count(0)=count(0)+1
      else if((d(j).gt.0.).and.(d(j).lt.1.).and.(i.eq.0)) then
          count(0)=count(0)+1
      else if ((d(j).ge.i).and.(d(j).lt.(i+1))) then
c
      if(i.eq.0) then
          go to 170
      end if
c
          count(i)=count(i)+1
      end if
c
170   continue
160   continue
c
c     ************************************************************************
c
c     the following section determines the spatial distribution
c     of total rainfall depth (fraction of total area wetted to a depth
c     greater than or equal to)
c
c     ************************************************************************
c
      totcount=0.0
      print,' '
      print,'the following is the spatial distribution of total rainfall depth'
      print,'..........................................'
      print,' '
      do 180 i=max+1,-2,-1
      if(i.eq.-2)then
      print1050,count(-2)
      else if(i.eq.-1) then
      print1060,count(-1)*.01
      else if((count(0).gt.0.).and.(max.eq.0).and.(i.eq.0))then
          totcount=totcount+count(0)
          totcnt2(i)=totcount
      print1070,totcount*.01
      else if((count(0).gt.0.).and.(i.eq.0)) then
          totcount=totcount+count(0)
          totcnt2(i)=totcount
      print1070,totcount*.01
      else if(max.eq.0) then
      totcount=0.0
          totcnt2(i)=totcount
      else
      totcount=totcount+count(i)
          totcnt2(i)=totcount
      print1080,i,totcount*.01
      end if
180   continue
c
c     ***********************************************************************
c
c     write the results to file93
c
c     ***********************************************************************
c
c
          e=k
          kk=amax+2
      write(93,1130)kk
      write(93,1140)iyear(1),imonth(1),iday(1)
      do 190 i=-1,amax
      if(i.eq.-1) then
      write(93,1090)count(-1)/e
          go to 190
      else if(i.eq.0) then
      write(93,1110)totcnt2(0)/e
      else
      write(93,1120)i,totcnt2(i)/e
      end if
          x(i+1)=i
          y(i+1)=totcnt2(i)/e
190   continue
c
1050  format('there were ',f9.0,' neg. elements in the integration routine')
1060  format('total unwetted area (0 mm. rainfall)=',f9.2,' sq.km.'/)
1080  format('area wetted by at least ',i3,' mm. or more was ',f9.2,' sq.km.')
1070  format('area wetted by more than 0 mm. was',f9.2,' sq.km.')
1090  format('dry fract. total area (0 mm. rainfall)=',f9.3)
1120  format('fract. of area wetted by at least ',i3,' mm. or more was ',f9.3)
1110  format('fract. of area wetted by more than 0 mm. was',f9.3)
1130  format('there are',i3,' lines in this file')
1140  format(3(i2))
1010  format('storm day ',i2,'/',i2,'/',i2)
c
      end
```

```
c                         TABLE.FORTRAN
c
c        The purpose of this program is to create nicely formatted
c        storm day data table.
c
c       This program was developed by Neil M. Fennessey at M.I.T. during
c       the course of his Master's Degree research about the Areal Distribution
c       of Rainfall
c
c        ..............................................................
c
        dimension   amonth(6)
        dimension   xarea(100),ygamma(100),lag1(100),corr1(100)
        dimension   wetfrct(-1:200)
c
        data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
c       read in the results from stormday.fortran, date, moments,
c       spatial correlation and variance function curves
c
        read (91,1160)iyear,imonth,iday
        read (91,1170)avedepth,vardepth,skwdpth
        do 10 i=1,31
        read (91,1180)lag1(i),corr1(i),xarea(i),ygamma(i)
10      continue
c
1160    format(3i2)
1170    format(3(1x,f7.3))
1180    format(f5.2,1x,f7.3,1x,f6.2,1x,f6.3)
c
c       read in the results of stormwet.fortran (spatial distribution curves)
c
        read(93,1130)kk
        read(93,1140)iyear,imonth,iday
        do 30 i=-1,kk-2
        if(i.eq.-1) then
        read(93,1090)wetfrct(i)
        else if(i.eq.0) then
        read(93,1270)wetfrct(i)
        else
        read(93,1120)i,wetfrct(i)
        end if
30      continue
            iiday=iday
            iiyear=1900+iyear
            iii=imonth-5
            15=iyear-20
c
c       write the table to file95
c
        write(95,1005)amonth(iii),iiday,iiyear
        write(95,1010)wetfrct(-1)
        write(95,1050)wetfrct(0)
        write(95,1290)avedepth
        write(95,1280)vardepth
        write(95,1285)skwdpth
        write(95,1055)
        write(95,1060)
        write(95,1070)
        write(95,1220)
```

```
c
c     check to see is the cumulative wetted area is larger or smaller
c     than the fixed number of observations of area correlation lag
c
      if(kk.gt.31) then
          k1=kk-2
          b=1
      else
          k1=31
          b=2
      end if
      do 40 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      write(95,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else if((b.eq.1).and.(i.gt.31)) then
      write(95,1240)i,wetfrct(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(kk-2))) then
      write(95,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else
      write(95,1250)lag1(i),corr1(i),xarea(i),ygamma(i)
      end if
40    continue
c
c     write a versian to file 02 for future analysis
c
      write(02,1370)n,iiyear
      write(02,1310)amonth(iii),iiday,iiyear
      write(02,1320)avedepth,vardepth,skwdpth
      write(02,1330)wetfrct(-1)
      write(02,1340)wetfrct(0)
      write(02,1300)kk-2,k1
      write(02,1350)
      write(02,1360)
      do 50 i=1,k1
c
c     cumulative wetted fraction is greater than 31 mm.
c
      if ((b.eq.1).and.(i.le.31)) then
      write(02,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else if((b.eq.1).and.(i.gt.31)) then
      write(02,1240)i,wetfrct(i)
c
c     cumulative wetted fraction is less than 31 mm.
c
      else if ((b.eq.2).and.(i.le.(kk-2))) then
      write(02,1230)i,wetfrct(i),lag1(i),corr1(i),xarea(i),ygamma(i)
      else
      write(02,1250)lag1(i),corr1(i),xarea(i),ygamma(i)
      end if
50    continue
c
c     file02 format statements
c
1370  format(' there are ',i5,' storm days in this file for this year:'i4)
```

```
1360  format(7x,'Y  (mm.) ',2x,'Acw/Ac  (Y>y) ',10x,'v  (km.) ',1x,'rho(v) ',8x,'A
     &(km.sq.) ',1x,'Gamma(A) ')
1350  format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
     &'Variance Function')
1340  format(' Wetted Fraction of Total Basin Area:  (Acw/Ac)=',f5.3)
1330  format(' Dry Fraction of Total Basin Area:  (Acd/Ac)=',f5.3)
1320  format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3,' S.C.(Y)=',f7.3 )
1310  format(' Storm Day  ',a4,i3,i5)
1300  format(' there are  ',i3,' wetted area curve pts, there are  ',i3,
     &' total data points this day')
c
c      file95 format statements
c
1285  format(10x,'Coef. of Skewness of Point Depth: S.C.(Y)=',f7.3//)
1280  format(10x,'Variance of Point Depth  (mm. sq.): Var(Y)=',f7.3/)
1290  format(10x,'Expected Value of Point Depth  (mm.): E(Y)=',f7.3/)
1250  format(37x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1240  format(7x,i3,7x,f5.3)
1230  format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,6x,f5.3)
1220  format('________________________________________________________________
     &______________'/)
1070  format(6x,'y(mm.) ',4x,'Acw/Ac(Y>y) ',10x,'v(km.) ',2x,'rho(v) ',6x,'A
     &(km.sq.) ',2x,'Gamma(A) ')
1060  format(6x,'of Total Storm Depth',8x,'Spatial Correlation',5x,
     &'Variance Function')
1055  format(6x,'Spatial Distribution')
1050  format(9x,'Wetted Fraction of Total Basin Area:  (Acw/Ac)=',f5.3//)
1010  format(10x,'Dry Fraction of Total Basin Area:  (Acd/Ac)=',f5.3/)
1005  format(25x,'Storm Day  ',a4,i3,i5///)
1090  format('dry fract. total area  (0 mm. rainfall)=',f9.3)
1120  format('fract. of area wetted by at least  ',i3,' mm. or more was  ',f9.3)
1270  format('fract. of area wetted by more than 0 mm. was',f9.3)
1130  format('there are',i3,' lines in this file')
1140  format(3(i2))
      print,'please shift file02 to the bottom of file',15
      end
```

```
c                            FUTURE.FORTRAN
c
c     the purpose of this program is read the stacked data the storm day
c     archive data files, arc70.data, arc71.data,...,arc77.data.
c      Data from a particular year may be accessed by this program
c     following suitable modification by the analyst.  In its present
c     form, the data for an entire year is simply read into active
c     memory and nothing else.
c
c     Prior to an execution,for the particular year, arc70.data must be
c     placed in file50, arc71.data placed in file51, and so forth
c
c     The data is tabulated by storm date and includes the expected value
c     of the point storm depth, variance of the point storm depth, the
c     fraction of the catchment area that is dry for that storm day, and
c     the fraction of the catchment area that is wetted (greater than 0.01 mm.)
c     The variance function values for different areas as well
c     as the correlation function values for different lags are
c     also included.( area equals km. squared, lags are in kilometers)
c     The spatial distributin of wetted area curve values are also given.
c     They are defined as that percentage of the basin area wetted to a
c     depth greater than or equal to the corresponding depth (given in mm.)
c
c     This program was developed by Neil M. Fennessey at M.I.T. during the
c     course of his Master's degree research about the Areal Distribution
c     of Rainfall
c
c     ..................................................................
c
c                          Definition of Variables
c
c                               Storm Day Date
c
c     iyear         storm day year
c     iyar          storm day year
c     imonth        storm day month
c     amonth        storm day month
c     iday          storm day calendar date
c     iiday         storm day calendar date
c
c                               Moments
c
c     avedepth       E(Y) of the storm day random field (mm.)
c     vardepth       VAR(Y) of the storm day random field (sq. mm.)
c
c                          Spatial Correlation
c
c     lag1          spatial lag distance (km.)
c     corr1         spatial correlation at lag1
c
c                          Variance Function
c
c     xarea         area over which Y has been averaged (sq. k.m)
c     ygamma        variance function gamma(area) at an area = xarea
c
c            Spatial Distribution of Total Rainfall Depth
c
c     dryarea       fraction of the catchment area left dry during the
c                             storm day precipitation event
c     wetarea       fraction of the catchment area wetted during the
```

```
c                                  storm day  precipitation event
c     wet           number of discrete data points in the spatial distribution
c                                  curve
c     tot           number of lines of discrete data for the spatial
c                                  distribution curve, the spatial correlation
c                                  curve, and the variance function curve
c                                  ( tot >= 31 )
c     idepth        depth (mm.)
c     wetfrct       fraction of catchment area wetted to a depth greater than
c                                  or equal to idepth
c
c     .............................................................
c
c
      dimension   imonth(70),iyear(70),iday(70),wet(70),tot(70),iyar(70)
      dimension   wetarea(70),dryarea(70),wetfrct(70,200),idepth(70,200)
c
      real lag1
c
c      data (amonth(i),i=1,5)/'June','July','Aug','Sept','Oct'/
c
      integer wet,tot
c
      common /fut1/xarea(70,31)
      common /fut2/ygamma(70,31)
      common /fut3/lag1(70,31)
      common /fut4/corr1(70,31)
c
      print,' enter last two digits of the year of interest:19__  '
      input,l1
          l2=l1-20
          l=l2
c
      rewind(l)
c
c     **************************************************************************
c
c     Preset the common block variables equal to zero
c
c     **************************************************************************
c
      do 20 i=1,70
      do 30 j=1,31
          xarea(i,j)=0.0
          ygamma(i,j)=0.0
          lag1(i,j)=0
          corr1(i,j)=0.0
 30   continue
 20   continue
c
c     **************************************************************************
c
c      check to see how many storm days are in this year's storm day archive
c      data file
c
c     **************************************************************************
c
       read(l,1370)n,iyr
c
       do 10 j=1,n
```

```
c
c     ***********************************************************************
c
c       read entire contents of this year's data archive file
c
c     ***********************************************************************
c
      read(1,1310)amonth,iday(j),iyar(j)
          iiday=iday(j)
          iyear(j)=iyar(j)-1900
          iii=imonth(j)-5
      read(1,1320)avedepth,vardepth
      read(1,1330)dryarea(j)
      read(1,1340)wetarea(j)
      read(1,1300)wet(j),tot(j)
      read(1,1350)
      read(1,1360)
c
          kk=wet(j)
          k2=tot(j)

c
c     ***********************************************************************
c
c      Determine if the total number of discrete values for this storm day's
c      spatial distribution curve is less than or exceeds the 31 discrete
c      values of the spatial correlation and variance functions (all three
c      are written on the same line of the individual storm day record)
c
c     ***********************************************************************
c
       if(kk.gt.31) then
           k1=k2
           b=1
       else
           k1=31
           b=2
       end if
       do 50 i=1,k1
c
c     ***********************************************************************
c
c     The number of discrete values in the spatial distribution curve is
c      less than or equal to the 31 discrete values of the spatial correlation
c      and variance function curves
c
c     ***********************************************************************
c
       if ((b.eq.1).and.(i.le.31)) then
       read(1,1230)idepth(j,i),wetfrct(j,i),lag1(j,i),corr1(j,i),xarea(j,i)
      &,ygamma(j,i)
c
c     ***********************************************************************
c
c     The number of discrete values in the spatial distribution curve is
c      greater than the 31 discrete values of the spatial correlation and
c      variance function curves
c
c     ***********************************************************************
c
```

```
      else if((b.eq.1).and.(i.gt.31)) then
      read(1,1240) idepth(j,i),wetfrct(j,i)
c
      else if ((b.eq.2).and.(i.le.kk)) then
      read(1,1230) idepth(j,i),wetfrct(j,i),lag1(j,i),corr1(j,i),xarea(j,i)
     &,ygamma(j,i)
      else if ((b.eq.2).and.(i.gt.kk)) then
      read(1,1250) lag1(j,i),corr1(j,i),xarea(j,i),ygamma(j,i)
      end if
50    continue
c
10    continue
c
c     ***************************************************************************
c
c      file__ format statements
c
c     ***************************************************************************
c
1370  format(' there are ',i5,' storm days in this file for this year:'i4)
1360  format(7x,'Y (mm.) ',2x,'Acw/Ac (Y>y) ',10x,'v (km.) ',1x,'rho(v) ',8x,'A
     &(km.sq.) ',1x,'Gamma(A) ')
1350  format(3x,'Cumulative Wetted Fraction',5x,'Spatial Correlation',5x,
     &'Variance Function')
1340  format(' Wetted Fraction of Total Basin Area: (Acw/Ac)=',f5.3)
1330  format(' Dry Fraction of Total Basin Area: (Acd/Ac)=',f5.3)
1320  format(' point depth E(Y)=',f7.3,' Var(Y)=',f7.3)
1310  format(' Storm Day ',a4,i3,i5)
1300  format(' there are ',i3,' wetted area curve pts, there are ',i3,
     &' total data points this day')
c
1250  format(37x,f3.1,5x,f5.3,7x,f6.2,7x,f5.3)
1240  format(7x,i3,7x,f5.3)
1230  format(7x,i3,7x,f5.3,15x,f3.1,5x,f5.3,7x,f6.2,7x,f5.3)
c
      end
```